# computing today No 3

January 79

THE NEW MAGAZINE
FOR SMALL SYSTEMS
WITH BIG IDEAS .......


# Personal Computing~ The Early Years

# Softspot

**Presented as a supplement to ETI.**

# Introducing the personal computer you've waited for. THE EXIDY SORCERER.

**SORCERER COMPUTER SYSTEM** *Complete with Monitor*

The Sorcerer Computer is a completely assembled and tested computer system. Standard configuration includes 63-key typewriter-style keyboard and 16-key numeric pad, Z80 processor, dual cassette I/O with remote computer control at 300 and 1200 baud data rates, RS232 serial I/O for communications, parallel port for direct Centronics printer attachment, 4K ROM operating system, 8K ROM Microsoft BASIC in Rom Pac™ cartridge, composite video of 64 char/line 30 line/screen, 128 upper/lower case ASCII set and 128 user-defined graphic symbols, operation manual, BASIC programming manual and cassette/video cables, connection for S-100 bus expansion.

**only £950** Credit facilities available. + 8% VAT

**LOOK!**
- *32K RAM on board
- *RS232 interface *8K BASIC ROM
- *CUTS interface *4K MONITOR
- *KANSAS CITY interface *S100 BUS
- *User defined graphic symbols *Z80 cpu

# KEY BOARD
**756 GEORGE RISK**

**Brand new professional ASCII keyboards (USA)**

Full technical details included. **RRP £60.00**

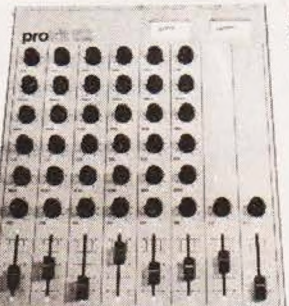
Only **£49.90** + 8% VAT.

Ready built, tested and guaranteed.

# COMPUTER JOYSTICK
Plugs into your Nascom P.I.O. No extras. Software and full documentation supplied. Plus free game cassette.

**£14.90** each **£28.90** per pair

# COMP PRO Mixer
**Professional audio mixer that you can build yourself and save over £100.**



**6 into 2** with full equalization and echo, cve and pan controls.

All you need for your own recording studio is a stereo tape or cassette recorder.

This superb mixer kit has slider faders, level meters and additional auxilliary inputs.

Only **£99.90** plus 8% VAT for **complete** kit Plus **FREE** power supply valued at £25.00

**Ideal for**
**DISCOS STAGE MIXING HOME STUDIOS AND MANY OTHER APPLICATIONS**

# COMPUTER COMPONENTS
Send for our Spring 1979 catalogue. **0.60p** Full of Computer Components, Peripherals and systems.

# Teleplay presents the PROGRAMAGAME of all time!

TELEPLAY have produced a new product of easy to assemble cartridge games, for you and all your family, that will not go out of date. Just ADD each new cartridge, as it becomes available, and your colour video centre will be up to date the Teleplay way. You will save pounds in the long run. Works OK on Black and White TV.

EX STOCK IMMEDIATE DELIVERY

REALISTIC SOUND EFFECTS

We offer all this to you for ONLY **£28·90** + £2.31 VAT
Fully Assembled £35.90

The Teleplay Kit consists of a Professional Finished Inject Moulded Box. Joystick Hand Controls – one cartridge which consists of 10 games Football, Tennis, Solo Squash (1 & 2 players), Hockey, Basketball (1 & 2 players), Shoot etc. Mains Adaptor. All holes are pre-drilled – No Special equipment required. You'll be surprised how easy it is to assemble this professional kit! Instruction Manual supplied separately at 0.45p. TELEPLAY GIVES YOU TWICE THE FUN BUILDING AND PLAYING THE MOST PROFESSIONAL TV GAME KIT EVER OFFERED.

Main Console has 3 Facility Handicaping Switches

Every 3 months a NEW game will become available to you and those already under production include Submarines, Space War, Wipe Out and many more. The price of these will vary from £12 to £19 depending upon the complexity of the game.

Additional cartridges NOW available include:

| | | | |
|---|---|---|---|
| Stunt Cycle | 4 games | Cartridges and Hand Controls | £14.90 + £1.14 VAT |
| Tank Battle | Cartridge and Hand Controls | | £18.90 + £1.51 VAT |
| Road Race | 2 games | Cartridge Controls already with basic kit | £12.90 + £1.03 VAT |

Fully assembled add £3.00

**NEW–**
**WIPE OUT cartridge – 6 games £13.90 + £1.11 VAT**

Cartridges come complete with printed case and easy to follow assembly instructions.

All parts fully guaranteed.

*Electrical knowledge is not a necessity to assemble this project – just simple soldering*

# INTERESTED IN HOME COMPUTING?

FREE B BUG valued at £23.00 plus 10 x C12 cassettes valued at £4.00 WITH EVERY NASCOM

Start now and don't get left behind THE NASCOM 1 is here **Ex-stock with full technical services**

Plus the opportunity to join the fastest moving club of personal computer users enabling you to get the most our of your computer. You can OBTAIN and EXCHANGE programs and other software – many now available.

The Powerful Z80 Microprocessor
Professional Keyboard
1 Kbyte Monitor in EPROM
2 Kbyte RAM (expandable)
Audio Cassette interface
Plugs into your domestic TV
Easy construction from straightforward instructions
– no drilling or special tools
– Just neat soldering required.

Only **£197.50** + 8% VAT (includes p & p + insurance)

| | | | |
|---|---|---|---|
| Manuals seperately | 2.95 | Monitor quality improved TV Modulator | 2.50 |
| Z80 programming Manual | 6.90 | | |
| Z80 Technical Manual | 2.95 | | |
| PIO Technical Manual | 2.95 | Power supply suitable for NASCOM | 19.90 |
| (All prices add 8% VAT) | | | |

**NASCOM AD ONS** – Nascom improved monitor B Bug (2K) featuring – *Four times tape speed *Direct text entry without ASCII *Extended keyboard facility *Additional useful subroutines **£23.00**

**Nascom Software library. Send SAE for lists and prices.**

**BLANK C12 Racal Quality CASSETTES £4.00 for 10**

All prices include VAT except where shown. Orders over £5 post and packing free otherwise add 20p. Please make cheques and postal orders payable to COMP, or phone your order quoting BARCLAYCARD or ACCESS number.

**COMP** COMPUTER COMPONENTS

**14 STATION ROAD · NEW BARNET · HERTFORDSHIRE · TEL: 01-441 2922 (Sales) 01-449 6596**
CLOSE TO NEW BARNET BR STATION – MOORGATE LINE
**OPEN – 10am to 7pm – Monday to Saturday**

# computing today
## No.3 January 1979



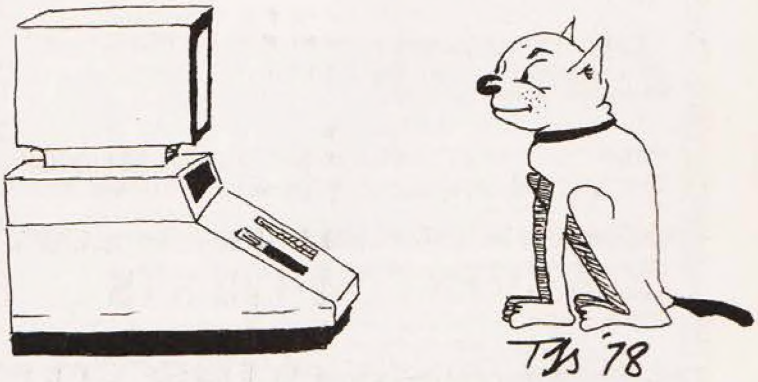


Editor: Gary Evans
Production: Pete Howells
Advertising: Mark Strathern, David Sinfield, Joy Cheshire


A number of errors that crept into last month's issue have been brought to our attention. Once again Phil Cornes was not credited in connection with the BASIC explained Series. Q1 in the EPROM Programmer should be shown as a 2N3638 and not PN3638 as in both Parts list and Circuit Diagram. The opto isolator is not a critical component and could be substituted by a more readily available device.

The missile program appearing in Softspot also contained a number of errors. Lines 90, 110 and 170 should be altered as follows

```
90 Y = (RND(16)—1)* 64;Z = 1
110 VDU Z,32
170 VDU @(I+2),32
```



**Man's best friends**

# Letters

**We would like to hear your views on computers and computing, we'll publish the best views in this, our new letters' page. This month, Mike Hughes — designer of the TRITON — answers a letter drawing attention to certain aspects of the computer design.**

*Dear Sir,*

*I am writing to request you to urgently consider the redesign of the Triton Computer, as it suffers from a serious design fault which will make it unreliable as it stands, and which will cause difficulties for expansion. As you may by now be aware I am referring to the data bus buffering. While the 8080a chip set has been buffered by a high drive (and unnecessary) 74LS245, the memory chips must also drive the data bus, and these have a single TTL drive capability. Bits 7 and 8 of the data bus are the most heavily loaded, with 5 LSTTL loads (= 1 TTL load), plus the UART, which presents 1 TTL load (at least this is what Texas say for the TMS 6011). Hence the memory chips are having to drive twice their rated load. Specifications being what they are this will usually be OK, especially when running so slowly. However usually is just not good enough in this situation, and several constructors are bound to get chips which are close to specification, and will be intermittent errors. The problem will become significantly worse when the bus expansion connector is used, as this will add an extra load, together with a lot of added cross-talk and bus noise.*

*As many of the people who may be considering the construction of this unit will not have the knowledge or equipment to detect this problem, I must, on their behalf, urge you modify this design as soon as possible.*

*A second, though less important, potential problem is in the -5V power supply, which is not a minor bias supply for 2708s (at least not according to Texas and Intel specifications). The specification is 30mA typical, 45mA max. Hence the power supply should be able to supply 180mA, which implies a series resistor of about 39-. This will dissipate a lot of power, and an IC regulator would be better.*

*Yours faithfully,*
*R. A. Cottis,*
*Corrosion and Protection Centre*
*UMIST*

I cannot disagree with anything in Mr Cottis' letter. I am, in particular, grateful to him for identifying the -5V rail problem which was an oversight on my part.

Unfortunately, the values given for Rl and ZD1 were a "hangover" from an early prototype which operated with a single EROM. These should have been changed to accommodate the higher current consumption of the completed system. Mr Cottis is quite correct in saying that Rl should be 39 ohm with a power rating of 1.5W. At the same time, ZD1 should be upgraded to have a power dissipation of 1W. The latter is necessary in the event of the system being operated with only one EROM in position.

Fortunately (or unfortunately as the case may be) this error will not show itself in cases which operate with 3 EROMS whose current consumption averages just below the "typical" value. Not many people will, therefore, have experienced any problem. If they had the error would have been discovered earlier. Readers who have their system up and running with 3 Eroms in position need not react to the problem urgently — nevertheless, they should upgrade these components in due course prior to inserting a 4th EROM. Constructors just about to start are strongly advised to use the higher rated components from the word go.

The driving capabilities of the 2111 memories on the data bus is a much more difficult question to answer. Mr Cottis is quite correct in every thing he says by taking worst case input loads and output drive capabilities. The worst case conditions he mentions are, however, stated with a 0.4 or 0.45V maximum "O" level — allowing a 400mV low level noise margin.

Stretching the loading in the worst case condition will certainly encroach on the noise margin but as $V_{OL}$ rises, as a result, so the input current will decrease. From figures available it would appear that in the worst case condition (including an extra LS load for an external bus driver), the noise margin could reduce to about 200 mV, and this would not be acceptable for stringent applications or in extremely high noise conditions.

As Mr Cottis implies, it is very unlikely that one will be so unlucky as to have all the worst case conditions stacked against him and this is what I was relying on to provide a reasonable noise margin for domestic/office noise environments. The "Bête Noire" is clearly the UART but I was reluctant to provide extra on board buffering for this for two reasons:-

1. It would have given rise to difficulties in layout — involving more board area.
2. This would have increased the board cost as well as possibly requiring extra components.

If one accepts that an extreme worst case conditions is unlikely there will be negligible problem at normal ambient temperatures. Even under a worse case situation at temperatures up to 60°C the system will still operate — but with impaired noise margin. It is in the latter situation that one might have problems with busbar noise etc.

I am indebted to Mr Cottis for drawing attention to this potential problem and in view of his comments, would suggest that readers planning expansions should keep their first umbilical cable from the main board to the peripheral mother board (soon to be published) as short as possible. They should also introduce a further bi-directional buffer on the data bus at the earliest position possible. This, incidentally, is already planned on the extension mother board.

To assure readers of the minimal chance of problems arising, we should point out that of the dozens of Tritons already built and working, there has not been one instance of bus noise problems. This includes one system which, already, has been externally expanded for a further 4k of RAM with no extra buffering.

M. J. Hughes

# TRS-80: A Small Business Application

TERRY JOHNSON owns, with a partner, an insurance brokerage firm with twelve branches spread across the country. The firm also has interests in various property. Having used a computer bureau to process various accounting information, the firm are at present installing a TRS-80 to provide all the bureau facilities plus additional services.

**How did you start in business?** About 20 years ago, after coming out of the RAF, where I was a night fighter pilot, I needed a job. I had no business background but managed to get a job with an insurance company. There I found out there were such things as insurance brokers and after a bit of a late night study, I set myself up as a broker. Things, like topsy, grew from there, at our peak in 1974 we had 18 branches, but have now cut back to 12 — We employ 10 managers and about 40 other staff.

**Your first involvement with computers was via a computer bureau?** That's right, about five years ago we decided that the volume of accounting work generated by the various branches was getting to a level where some form of automation was necessary. We approached a number of the recognised agencies to see if they could offer us the type of package we wanted, without exception the products they offered would have meant considerable changes to our systems — not at all what we wanted.

At this point we decided on a different approach, we found — through the yellow pages I think—a specialist computer firm which undertook to design programs to customers' requirements. The firm had a scientific background and had never tackled anything along the lines we were asking. They accepted the challenge and a short time later we had a FORTRAN program that did all we wanted — simply keeping track of money in and money out and of our commodity — insurance policies in our case — but it might just as easily have been nuts and bolts or oranges.

**Did you encounter any problems with the system?** No, everything went very smoothly, we prepared our own program cards on a second hand flexowriter we picked up for £1000 and 48 hours later we had our batch back. During that first year about the only thing that happened was that the system was improved to provide a profit/loss statement and balance sheet.

**When did the Tandy TRS-80 come onto the scene?** About the middle of the year but the first time I realised that an "in house" system might be within our reach was at the DIY computer show at the beginning of this year.

**Why even consider a new system when your existing one was working so well?** To save money. We calculated that a system costing around £2500/£3000 could save us money over a period of 5 years with, of course, all the benefits of in house computing.

**Back to the Tandy then?** — Yes, in April of this year I went to the States on holiday and after looking around at what was available off the shelf over there at that time I decided to buy a Tandy TRS-80. The machine I bought was a Level 1 16K machine.

**Any problems getting it through customs?** Not after it had been classified as a data recording machine by binary system — no I paid my 7% duty on hardware and 8% VAT and I was through. I'd saved myself a lot of money. The cost in pounds was about £520. compared to the £700 odd that Tandy UK wanted assuming there was such a machine in this country.

**What about servicing if the machine goes wrong?** Tandy have a 90 day warranty on their equipment, which is available to first and subsequent owners in any country where Tandy have outlets — I wasn't worried about servicing.

**Have you had any trouble with your machine?** Yes, soon after I began using it, the keyboard suffered from an excessive amount of contact bounce. I took it to my local Tandy dealer, and it was back within a week — repaired at no charge under the warranty.

**How did you get on with the machine in those early days?** I had no experience of programming but found the Tandy manual soon had me familiar with BASIC and frustrated at the limitations of the level 1 machine.

I sent my TRS-80 to Tandy together with £79 for it to be upgraded to a Level 2 unit.

**What was your next step?** Well now came the most frustrating time for me, Level 2 BASIC was just what I wanted, but without a printer, floppy and more

memory the system just could not cope with the work I wanted it to do. Tandy had plans for expansion but no hardware available. I started looking around at other machines. The PET would never be acceptable to any typist with its present keyboard but the APPLE looked promising. I bought an apple this July but returned it about a week later, the reasons were that my machine kept crashing and as the BASIC I was using, Applesoft II was on tape, a five minute restart each time became trying. Add to this the fact that I was promised a number of hardware items that just didn't turn up and the Apple was not working out.

**Back to Tandy?** Yes. August was a dull month but I'd been promised the first expansion interface to arrive in this country and that arrived in September. The expansion interface contains space for an additional 16K or 32K RAM, a dick-controller for up to four disks, dual cassette decks as well as a Centronics parallel port.

After a bit of persuasion Sintrom at Reading — having sold me their ex-demonstration printer — hooked the printer up to the TRS-80, as the lead supplied did not produce any results this was a great leap forward — hard copy.

Things were moving fast now, as soon after this I discovered that a Micropolis disc drive would plug directly into the Tandy interface. I now had a system that could do all that I demanded of it.

**Does this bring us up to date?** No, a couple of weeks ago I went to the States again and picked up a couple of Tandy disks, a screen printer, a back up CPU and some extra memory. About $3000 worth, again a considerable saving on buying over here.

**What about the software development?** Of course that's been going on all the time, I've had no real difficulties with the task. The bulk of the work is done now, it's just a matter of getting the time to sort out the rest of the system for our target start date of January 1st.

**Why should a small business man consider using one of these small computer systems?** Three main reasons I think. Staff time and skill levels can be cut if a sensible system is devised. Accounting information that is good enough for our auditors to accept can be produced. Thirdly a vast flow of statistical information can be generated that gives the business man a far better idea of the performance of his operation at any point in time. What most people do by feel now can be done far more accurately by the computer. These three things together all add up to cutting costs.

**Do you think small business men could cope with setting up a system such as yours without outside help?** Most small businessmen are of above average intelligence — they have to be to survive. They also, in general have the drive and energy to get things done. I hadn't any knowledge of computers at the start of this year and reckon that now I have put together a system that will save me time and money. If I can do it I'm sure many other people in my position could. With the various software packages coming on the market at low cost, the business man may only be involved in a very small amount of work to get a low cost system to do exactly what he wants of it.

# TRITON Software —MONITOR

**Mike Hughes takes a detailed look at the Tritons monitor and describes some machine code programs that can be run on the computer**

If you have built the TRITON computer you will want to get the best out of it and there are several modes in which it can be used. You can write and run programs in BASIC but you may wish to record and recall these to and from tape. To do this you need to know something about the MONITOR. You may wish to use the computer for control purposes for which you need to write machine language instructions. To do this it is essential to go through the Monitor — possibly making use of some of its in-built sub routines.

For those learning about computers it is a very good idea to get a grasp of MACHINE CODE. It is not difficult to learn and can make life much more exciting as you will be able to get the computer to respond much more quickly than it will through a BASIC Interpreter. Sometimes you can get several thousand times the sped for certain operations. To do this on the TRITON you must operate through the MONITOR.

TRITON's standard MONITOR is a program written in machine code which is held in ROM starting at address location OOOOH. Its purpose is primarily to initialise the machine and to give it an elementary intelligence so that you can communicate with it. For this reason the machine must start off at the beginning of the monitor program every time it is switched on otherwise it becomes a worthless heap of rather expensive electronic components — unable to do anything. The first machine code instruction that the computer sees in the Monitor sets the STACK without which it would be impossible to do much in the way of decision making via nested sub-routines. It then proceeds to look at the next instruction which, in the case of TRITON, enables the interrupt operation if ever it is needed to be used. The following instruction is a JUMP which leads on to the first of the main routines called SCANMEM. This routine points to address 1600H and writes FFH into it. It then reads back the value from that address and checks that the FFH was actually stored. Furthermore it writes OOH into the same location and checks that too. This is a check that the memory is there and working which, at the same time leaves the memory location clear (i.e. containing OOH). The monitor routine then steps up one memory byte — to 1601H and does the same. This process is carried out on every successive location from 1600H upwards until the computer finds an error. The address of the location where the error occurred is most likely to be the top of the RAM work area but could be the address of a faulty IC. In either event this top address is written into a pair of RAM bytes used by BASIC 14.1 to tell BASIC how much work space is available. The two bytes of memory used for this are 1481H and 1482H. As a general rule whenever two byte instructions or data are written into memory the 8080 microprocessor expects to see the least significant byte in the lower value of the two addresses. Thus if 2000H was the location where memory ended — as found by SCANMEM, location 1481H would contain OOH while 1482H would contain 20H.

After checking the memory the monitor initialises the computer which acknowledges this on the screen with its standard message:

TRITON READY
FUNCTION? P G I O L W T

The Monitor then goes into a keyboard loop and the computer effectively waits for you to tell it what to do. This will depend on which key you depress. The letters it expects you to type are those shown in its acknowledgement and are abbreviations for seven different primary operations you can do with the monitor. These are:

P = Inspect any memory location and, if necessary change, or insert a byte of data. When the data is entered the computer automatically steps to the next address showing what is currently there and waits to see if you wish to change it.

G = Start running a program from any specified starting address location. The computer asks you, within this routine, what start location you want.

I = Input from tape recorder. The computer asks you for the header code of the file and then searches for it. When it has been found the data is written into the computer's memory starting at location 1600H. When the flag marking the end of the record has been found the computer re-initialises with an abbreviated form of its initial "switch on" message.

O = Output a program to the tape recorder. This, again asks you to give your recording a header code. The routine automatically outputs programs written in BASIC and stops when it gets to the end of file address (the address written into bytes 1600H and 1601H by BASIC). For user written machine code programs you have to manually enter the address immediately following your last instruction into these two bytes. Tapes are ALWAYS loaded and dumped with 1600H being the start location! When dumping has finished the computer re-initialises.

L = List the machine code content of all locations starting from any specified address. The

computer asks for the first address then prints out the contents of this and the next 14. It then asks MORE? and expects you to type Y otherwise it re-initialises.

W = Typewriter mode. The computer behaves just as if it was a keyboard and VDU. Anything you type is displayed on the screen including graphics. Cursor control and special VDU functions — e.g. Clear Screen, Reset Cursor etc all operate but the computer responds to nothing except CONTROL C which makes it re-initialise.

T = Jump to BASIC L4.1. This command causes the computer to jump out of the control of the Monitor into the control of BASIC. CONTROL C will jump back out of BASIC into the initialisation condition of the Monitor.

Note that CONTROL C will, in nearly all cases, get you out of an operation and back to the initialisation condition. The only times when it fails to do this are when you are locked in a user written machine code program loop; are searching tape for a non-existent header or are outputting to the tape recorder. In these three cases you will have to use Interrupt 2 (which re-initialises without clearing memory) or RESET which goes through the SCANMEM routine and erases any data in memory.

When the computer asks for its initial instruction via one of the above letters you simply have to type the letter. No carriage return is needed. If you type the wrong letter the computer replies "INVALID" and waits for you to try again.

Here are some examples to try with the above functions:

**LED PORT TEST ROUTINE**

| | | | | |
|---|---|---|---|---|
| 1600 | CALL INCH | | CD | Input data from keyboard to accumulator |
| 1601 | — | | 0B | |
| 1602 | — | | 00 | |
| 1603 | CMA | | 2F | Complement contents of accumulator. |
| 1604 | OUT | PORT 03H | D3 | Output contents of accumulator to |
| 1605 | — | | 03 | PORT 3 (LED port) |
| 1606 | JMP | 1600H | C3 | Jump back to 1600H for next input |
| 1607 | — | | 00 | from keyboard. |
| 1608 | — | | 16 | |

**This program enables you to test out both the LED port and the keyboard by outputting the binary code from the keyboard to the LEDs. The program complements the accumulator to compensate for the fact that on TRITON the LEDs go on for level "0." In this program they will go on if a bit from the keyboard is "1." Because we are going through the monitor's INCH routine you will find that a shift inversion takes place. You get upper case alpha codes when unshifted and lower case alpha codes when shifted. This is designed in to make the keyboard more convenient to use. Numerical keys are not affected by the shift inversion. Again, because we are using INCH you can escape from this program loop with CONTROL C. You will notice that the most significant LED (bit 8) is permanently off; this is because bit 8 is used for the input strobe and this bit is masked off by the monitor as data is entered. Notice also that the data is latched on to the LEDs after the key has been released.**

**INTERRUPT DEMONSTRATION PROGRAM**

| | | | | |
|---|---|---|---|---|
| 1618 | LXI D | 161FH | 11 | Load start of string saying |
| 1619 | — | | 1F | IAM INTERUPT 3 |
| 161A | — | | 16 | |
| 161B | CALL | PSTRNG | CD | Print carriage return followed |
| 161C | — | | 2B | by string. |
| 161D | — | | 00 | |
| 161E | RET | | C9 | Return to main program. |
| 161F | DATA | I | 49 | String data starts here and |
| 1620 | — | SPACE | 20 | terminates with end of text |
| 1621 | — | A | 41 | marker 04. |
| 1622 | — | M | 4D | |
| 1623 | — | SPACE | 20 | |
| 1624 | — | I | 49 | |
| 1625 | — | N | 4E | |
| 1626 | — | T | 54 | |
| 1627 | — | E | 45 | |
| 1628 | — | R | 52 | |
| 1629 | — | R | 52 | |
| 162A | — | U | 55 | |
| 162B | — | P | 50 | |
| 162C | — | T | 54 | |
| 162D | — | SPACE | 20 | |
| 162E | — | 3 | 33 | |
| 162F | — | EOT | 04 | |

**An example of a user written interupt routine using INT3 push button. Note that it starts at the re-vectored start address 1618:**

**Load this program then reinitialise and press INT3 button. Next press W and do a bit of screen typing and press INT3 from time to time. The interupt should announce its presence. We hope that this simple example will show you that, contrary to popular belief, interupts are quite easy to write programs for.**

**VIDEO TYPEWRITER PROGRAM**

| | | | | |
|---|---|---|---|---|
| 1600 | LXI D | 1612H | 11 | Load start address of message string |
| 1601 | — | | 12 | in DE register pair. This is |
| 1602 | — | | 16 | required by PSTRNG sub-routine. |
| 1603 | CALL | PSTRNG | CD | Call sub-routine which prints string |
| 1604 | — | | 2B | starting at address held in DE register pair. |
| 1605 | — | | 00 | |
| 1606 | CALL | PCRLF | CD | Call sub-routine which prints out |
| 1607 | — | | 33 | carriage return and line feed. . |
| 1608 | — | | 00 | |
| 1609 | CALL | INCH | CD | Call sub-routine which inputs data |
| 160A | — | | 0B | from keyboard and holds this in |
| 160B | — | | 00 | accumulator. |
| 160C | CALL | OUTCH | CD | Call sub-routine which prints out ASCII character from |
| 160D | — | | 13 | data in accumulator. |
| 160E | — | | 00 | |
| 160F | JMP | 1609H | C3 | Jump back to address 1609H and wait |
| 1610 | — | | 09 | for next character from keyboard |
| 1611 | — | | 16 | |
| 1612 | DATA | 0 | 4F | Start address of message string |
| 1613 | | / | 2F | |
| 1614 | | K | 4B | |
| 1615 | | SPACE | 20 | |
| 1616 | | T | 54 | |
| 1617 | | Y | 59 | |
| 1618 | | P | 50 | |
| 1619 | | E | 45 | |
| 161A | | ! | 21 | |
| 161B | | EOT | 04 | End of text terminator code. |

**When you run this program it acknowledges with the message O/K TYPE! and you can then use the TRITON as if it were in the W function operating mode (i.e. it becomes nothing more than a video typewriter). You can escape from the program by depressing CONTROL C (this applies to any program which repeatedly goes through the INCH routine).**

ALPHABET TWELVE TIMES OVER USING I/O

| | | | | |
|---|---|---|---|---|
| 1600 | RST1 | | CF | Clear screen via special restart instruction |
| 1601 | MVI B | 0CH | 06 | Set decimal value 12 (oCH) in B to specify |
| 1602 | — | | 0C | number of alphabets required. |
| 1603 | MVI A | 41H | 3E | Set ASCII code for "A" in accumulator. |
| 1604 | — | | 41 | |
| 1605 | CALL | OUTCH | CD | Print contents of accumulator. |
| 1606 | — | | 13 | |
| 1607 | — | | 00 | |
| 1608 | INR A | | 3C | Increment ASCII code in accumulator by one. |
| 1609 | CPI | 5BH | FE | Compare it with ASCII code which is |
| 160A | — | | 5B | one greater than Z. |
| 160B | JNZ | 1605H | C2 | If not greater than Z jump back to 1605H |
| 160C | — | | 05 | and repeat until complete alphabet is |
| 160D | — | | 16 | printed. |
| 160E | CALL | PCRLF | CD | If alphabet is completed output carriage |
| 160F | — | | 33 | return and line feed. |
| 1610 | — | | 00 | |
| 1611 | DCR B | | 05 | Decrement value in B register by one. |
| 1612 | JNZ | 1603H | C2 | If it is not zero we do not have 12 alphabets so jump back to |
| 1613 | — | | 03 | 1603H and repeat. |
| 1614 | — | | 16 | |
| 1615 | JMP | REINIT | C3 | If it is zero re-initialise. |
| 1616 | — | | B9 | |
| 1617 | — | | 02 | |

**This program should be compared with the one following as they both do the same thing — print the alphabet twelve times and then re-initialise. This, first, program uses conventional I/O techniques whereas the second makes use of the powerful memory mapped option to the TRITON's VDU. We hope you will recognise the tremendous difference in speed of operation between the two methods. Note that in both programs we make use of the RST1 instruction at the beginning. This is one of 8 special re-start instructions which are fixed destination CALL instructions. Using RST1 will call the sub routine at location 0008H which, in the case of TRITON's MONITOR is then re-vectored with a jump to 0134H. The routine in question is the one which clears the VDU screen and resets the cursor. The advantage of using an RST instruction is that you do not have to specify the address of the sub routine hence saving two bytes in your program. You should only use RST instructions if the sub routine being called terminates in a RETURN command.**

**When you run the program try and judge the time it takes to display the 12 alphabets and then go on to the next example.**

ALPHABET TWELVE TIMES OVER USING MEMORY MAPPING

| | | | | |
|---|---|---|---|---|
| 1600 | RST1 | | CF | Clear screen with special restart instruction |
| 1601 | LXI H | | 21 | Load HL register pair with address one less than start of |
| 1602 | — | | FF | VDU RAM. |
| 1603 | — | | 0F | |
| 1604 | MVI B | 0CH | 06 | Set number of alphabets required in register B. |
| 1605 | — | | 0C | |
| 1606 | MVI A | 41H | 3E | Set ASCII code for "A" in accumulator. |
| 1607 | — | | 41 | |
| 1608 | INX H | | 23 | Increment HL register pair by one. |
| 1609 | MOV M,A | | 77 | Copy contents of accumulator to memory. |
| 160A | INR A | | 3C | Increment contents of accumulator. |
| 160B | CPI | 5BH | FE | Compare contents of accumulator with code |
| 160C | — | | 5B | one greater than Z. |
| 160D | JNZ | 1608H | C2 | If it isn't greater than Z jump back to 1608H and repeat. |
| 160E | — | | 08 | |
| 160F | — | | 16 | |
| 1610 | MOV A,L | | 7D | If it is: copy contents of L to accumulator |
| 1611 | ADI | 26H | C6 | Add 26H to this value. |
| 1612 | — | | 26 | |
| 1613 | CC | 1629H | DC | If addition causes a carry; call sub routine which |
| 1614 | — | | 29 | increments H register by one |
| 1615 | — | | 16 | |
| 1616 | MOV L,A | | 6F | Replace new low byte address in register L. |
| 1617 | DCR B | | 05 | Decrement register B by one. |
| 1618 | JNZ | 1606H | C2 | If it's not zero we do not have 12 |
| 1619 | — | | 06 | alphabets so jump back to 1606H and repeat. |
| 161A | — | | 16 | |
| 161B | MVI B | 0CH | 06 | Set decimal value 12 into B register. |
| 161C | — | | 0C | |
| 161D | MVI A | 0AH | 3E | Set accumulator to ASCII code for line |
| 161E | — | | 0A | feed. |
| 161F | CALL | OUTCH | CD | Output line feed to VDU to step cursor down. (to get it |
| 1620 | — | | 13 | clear of last alphabet). |
| 1621 | — | | 00 | |
| 1622 | DCR B | | 05 | Decrement register B by one. |
| 1623 | JNZ | 161FH | C2 | If it's not zero we have not stepped cursor to below the |
| 1624 | — | | 1F | last line of alphabet so jump back to |
| 1625 | — | | 16 | |
| 161FH and repeat | | | | |
| 1626 | JMP | REINIT | C3 | If it is; re-initialise. |
| 1627 | — | | B9 | |
| 1628 | — | | 02 | |
| 1629 | INR H | | 24 | Sub-routine to increment register H in event of a carry. |
| 162A | RET | | C9 | |

**Although this program is longer than the one just described you will see an element of similarity in the way the alphabet is formed (by incrementing the accumulator) and we keep track of the number of alphabets in the B register. Instead of using the I/O OUTCH routine we use the HL register pair to point to memory locations which are within the block of the VDU's RAM. This starts at 1000H and finishes at 13FFH. We then use the MOV M, A instruction which copies whatever is in the accumulator to the memory location being addressed by the HL register pair. By using the INX H instruction we can increment the latter to display the next character etc. Notice that carriage returns and line feeds are not needed in the main body of the program because we are using addressing to tell the computer exactly where to place each character. When one alphabet is finished we have to compute the address of the start of the next by adding the hex number 26 to the address currently in the HL register pair (this is done at instruction 1611H). Notice that after this operation we have to make allowances for a carry by calling an INR H sub routine.**

**When memory mapping the VDU you must remember that the clear screen/reset cursor operations does more than is immediately apparent to the eye. The addresses of different**

positions on the VDU screen must correspond to specific places on the screen (1000H is the top left hand corner and 13FFH is the bottom right corner). If, as a result of previous activity, the VDU screen has been scrolling these absolute address values do not correspond to positions in an absolute manner you must carry out a cursor reset operation. This can be on its own or combined with the screen clear operation. A similar operation must be carried out if you use the VDU function when under the control of BASIC L4.1. Note that it takes the VDU 132mS to carry out a home cursor operation so if every you output the raw instruction in your own software you must introduce a time delay greater than this before outputting anything else to the VDU. The Clear Screen/Reset Cursor utility in the monitor has a delay of about 200mS built into it so you can call it without introducing any further delay.

Note that when this program runs the cursor stays stationary in its reset position (top left corner of screen). If we allowed it to stay there the re-initialisation message would overprint our alphabets so we have included some extra instructions (starting at 161BH) which step the cursor down twelve positions immediately prior to re-initialisation.

To further demonstrate the flexibility of memory mapping you can alter the layout on the screen by altering the value added to the address to get the next line. Alter the data at location 1612H to 29 and re-run the program. The lines should have start points staggered by 3 character positions. Similarly you can alter the start location for the display as a whole by altering the data in locations 1602H and 1603H. Try making these 20 and 10 respectively and alter the data at 1612H to 25. When you run the program the rows should slant the other way.

When memory mapping the VDU you must always be careful to ensure that your memory pointer (HL register pair) cannot exceed the highest address of VDU (13FFH) otherwise you will start over-writing the input buffer and stack area of the monitor. If this happens all sorts of strange things will begin to take place and you will probably find you completely lose control through the keyboard. If this should happen you must resort to the RESET button and end up with a cleared memory!

# MEMORY MAP AND MONITOR UTILITIES

To help those who wish to get involved in machine code programming at an early stage here are the addresses of memory blocks and ports. It is assumed that you will be operating under the control of the standard Monitor program so we also list the addresses of its main Utilities.

## MEMORY START ADDRESSES

| | |
|---|---|
| 0000H - 03FFH 1K EROM | (Holds standard MONITOR) |
| 0400H - 07FFH 1K EROM | (Holds BASIC L4.1 "A") |
| 0800H - 0BFFH 1K EROM | (Holds BASIC L4.1 "B") |
| 0C00H - 0FFFH 1K EROM | (Spare location) |
| 1000H - 13FFH 1K RAM | (VDU memory — can only be written into by computer) |
| 1400H - 14FFH ¼K RAM | (Holds stack and tables for Monitor from 1400H to 147FH; 1480H upwards through this block will be used by BASIC L4.1 tables otherwise is free as M/C code work area) |
| 1500H - 15FFH ¼K RAM | (Completely reserved for BASIC L4.1 stack otherwise is free for the user) |
| 1600H - 1FFFH 2½K RAM | (Work area for BASIC L4.1 or user programs. NOTE that locations 1600H and 1601H are made use of by tape I/O routines to store End of File address — hence user programs for saving/loading to and from tape should always start at 1602H) |
| 2000H - FFFFH 56K | Available for "Off Board" extensions |

## PORT DESIGNATIONS

| | |
|---|---|
| 00H | Keyboard INPUT (NOTE special routine is necessary) |
| 01H | Tape I/O UART Status INPUT |
| 02H | Tape I/O UART Data Strobe (start transmission) OUTPUT |
| 03H | LEDs OUTPUT (NOTE LEDs are on for "O") |
| 04H | Tape I/O UART Receive Data Enable (receive data) INPUT |
| 05H | VDU OUTPUT (NOTE strobe — bit 8 — has to be specially formatted by software) |
| 06H | Spare OUTPUT (NOTE only bits 7 and 8 are output) |
| 07H | Relay OUTPUT (NOTE bit 8 is used to drive tape control relay out bit 7 is output as a spare line) |
| 08H - FFH | Available for "Off Board" extensions |

## MONITOR UTILITIES ETC

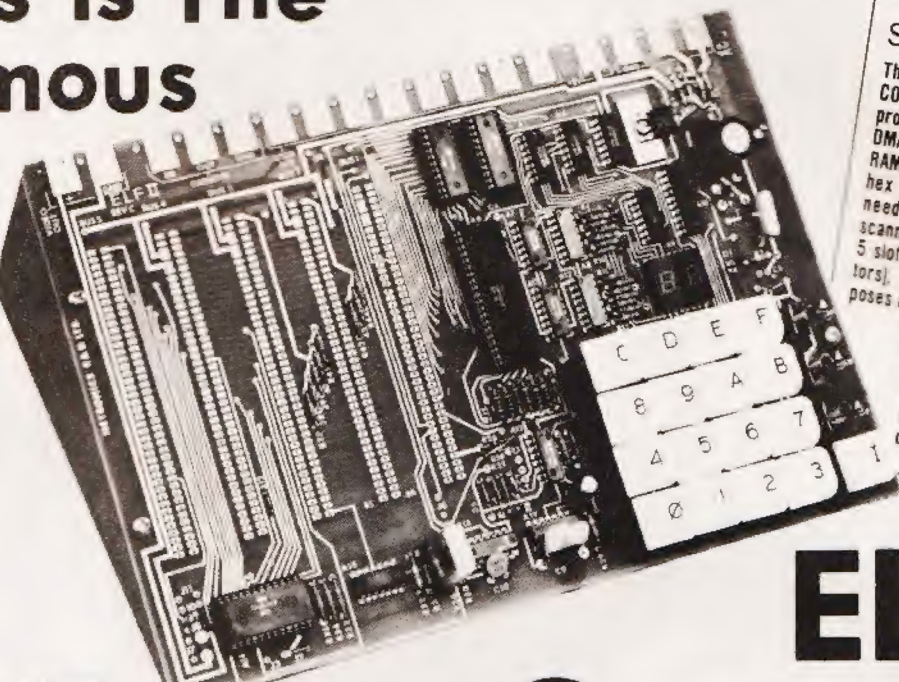
| | | |
|---|---|---|
| 0000H | RST0 | Reset address. Enables Interrupt and checks and clears memory writing number of bytes available into locations 1481H (low order byte) and 1482 (high order byte). With full main board memory in place this should read 2000H. Then goes on to Initialise computer. |

| | | |
|---|---|---|
| 0008H | RST1 | Reacts to Interrupt 1 clearing VDU and resetting cursor. |
| 000BH | INCH | A CALL routine which inputs character from keyboard. Stays in keyboard loop until key is depressed. |
| 0010H | RST2 | Reacts to Interrupt 2 clearing VDU and initialising computer without clearing memory. Used for normal reset operation. |
| 0013H | OUTCH | A CALL routine which outputs character to VDU. |
| 0018H | RST3 | Reacts to Interrupt 3 and re-vectors to 1618H for user routine. |
| 001BH | INDATA | A CALL routine which allows string of characters to be entered to any place in memory. Start location of string pointed to by DE register pair which must be pre-loaded before calling INDATA. Returns on Carriage Return. |
| 0020H | RST4 | Reacts to Interrupt 4 and re-vectors to 1620H for user routine. |
| 0023H | PDATA | A CALL routine which allows string of characters to be printed from any place in memory. DE register pair must be pre-loaded with start address before calling. Routine returns when it sees EOT terminator (04H) but DE steps to one address beyond terminator allowing immediate re-call for further string to be printed. |
| 0028H | RST5 | Reacts to Interrupt 5 and re-vectors to 1628H for user routine. |
| 002BH | PSTRNG | A CALL routine identical to PDATA except that it outputs Carriage Return/Line Feed prior to printing string. |
| 0030H | RST6 | Reacts to Interrupt 6 and re-vectors to 1630H for user routine. |
| 0033H | PCRLF | A CALL routine which prints Carriage Return followed by Line Feed and then returns with the original contents of accumulator intact. |
| 0038H | RST7 | Reacts to Interrupt 7 and re-vectors to 1638H for user routine. |

The above are fixed location utilities whereas the following are within the main body of the MONITOR which are liable to be re-located if new generation monitors are written. They are, nonetheless, very useful and can save a lot of redundant instructions elsewhere. NOTE you can only guarantee using these if you have MONITOR V4.1 on board!

| | | |
|---|---|---|
| 0053H | INIT | Destination of Interrupt 2. Clears screen and initialises computer. Also resets stack so should only be used as destination of a JUMP instruction. |
| 02B9H | REINIT | This should only be used as a JUMP destination. It resets the stack and re-initialises the computer WITHOUT clearing the screen. A useful instruction at the end of user programs to avoid having to use the HLT instruction. |
| 0327H | TPEON | When called switches the tape control relay on (relay contacts could be used for other control purposes by the user). NOTE that routine returns with 80H in accumulator. |
| 032CH | TPEOFF | When called switches the tape control relay off. Routine returns with 00H in accumulator. |
| 03A1H | ACKA | Start address of string which prints INVALID. Use should be made of PSTRNG utility to insert Carriage Return and Line Feed. |
| 03B5H | ACKB | Start address of string which prints START=. (Use via PSTRNG). |
| 03C4H | ACKC | Start address of string which prints HEADER=. (Use via PSTRNG) |
| 03CFH | ACKD | Start address of string which prints END. (Use via PSTRNG) |
| 00AAH | BASICIN | Checks to see if a key has been depressed on keyboard. If it has it returns with the data otherwise returns immediately with the accumulator cleared and flags set accordingly. Useful for real time interactive programs. |
| 010DH | DLY | When called introduces approximately 3mS time delay and returns with all registers (except flags) intact. |
| 0116H | FIVSEC | When called introduces a delay of between 5 and 6 seconds. Returns with registers and flags intact. |
| 0134H | CLRSCN | Clears screen and resets cursor. Approximately 200mS time delay is built into this routine. Returns with registers and flags intact. |

# This Is The Famous



## SPECIFICATIONS

The £99.95 ELF II computer features an RCA COSMAC COS/MOS 1802 8-bit microprocessor addressable to 64k bytes with DMA, interrupt, 16 registers, ALU, 256 byte RAM expandable to 64K bytes, professional hex keyboard fully decoded so there's no need to waste memory with keyboard scanning circuits, built-in power regulator, 5 slot plug-in expansion bus (less connectors), stable crystal clock for timing purposes and a double-sided, plated-through pc board plus RCA 1861 video IC to display any segment of memory on a video monitor or TV screen along with all the logic and support circuitry you need to learn every one of the RCA 1802's capabilities.

# ELF 11 Computer

# £99.95

**PLUS 8% V.A.T.**

Stop reading about computers and get your hands on one. With ELF 11 and our new Short Course by Tom Pittman, you can master computers in no time at all! ELF 11 demonstrates all 91 commands an RCA 1802 can execute and the Short Course quickly teaches you how to use each of the 1802's capabilities.

ELF 11's video output lets you display an alphanumeric readout or graphics on any T.V. screen or video monitor and enjoy the latest T.V. games.

But that's not all: once you've mastered computer fundamentals ELF 11 can give you POWER with add-ons that are among the most advanced found anywhere. American IEEE chapters plus hundreds of universities and major corporations have chosen the ELF 11 to introduce their students and personnel to microprocessor computing!

Learn The Skill That May Soon Be Far
More Important Than Your College Degree!

The ability to use a computer may soon be more important to your earning power than college degree. Without a knowledge of computers you are always at the mercy of others when it comes to solving highly complex business engineering, industrial and scientific problems. People who understand computers can command MONEY and to get in on the action, you must learn computers. Otherwise you will be left behind.

ELF 11 is the F-A-S-T Way to Learn
Computer Fundamentals!

Regardless of how minimal your computer background is now, you can learn to programme a computer in almost no time at all. That's because Netronics has developed a special Short Course on Microprocessor And Computer Programming in non-technical language that leads you through every one of the RCA COSMAC 1802's capabilities so you'll understand everything ELF 11 can do . . . and how to get ELF 11 to do it!

All 91 commands that an 1802 can execute are explained to you, step-by-step. The text, written for Netronics by Tom Pittman, is a tremendous advance over every other programming book in print.

Keyed specifically to the ELF 11, it's loaded with "hands on" illustrations. When you're finished, ELF 11 and the 1802 will no longer hold any mysteries for you.

In fact, not only will you be able to use a personal computer creatively, you'll also be be able to understand computing articles in the technical press.

If you work with large computers, ELF 11 and our Short Course will help you to understand what makes them tick.

A DYNAMITE PACKAGE FOR JUST £99.95 plus 8% VAT!

With ELF 11, you learn to use machine language — the fundamental language of all computers. Higher level languages such as FORTRAN and BASIC must be translated into machine language before a computer can understand them. With ELF 11 you build a solid foundation in computers so you really know what you're doing no matter how complicated things get.

Video output also makes ELF 11 unique among computers selling for such a low price. Attached to your T.V. set, ELF 11 becomes a fabulous home entertainments centre. It's capable of providing endless hours of fun for both adults and children of all ages! ELF 11 can create graphics, alphanumeric displays and fantastic video games.

Only a low cost R.F. Modulator is required to connect ELF 11 to your T.V.'s aerial socket! (To order one see coupon below.)

ELF 11's 5-card expansion bus (connectors not included) allows you to expand ELF 11 as your need for power grows. If you're an engineer or hobbyist you can also use ELF 11 as a counter, alarm, lock, thermostat, timer or for countless other applications.

## ELF 11 EXPLODES INTO A GIANT!

Thanks to ongoing work by RCA and Netronics, ELF 11 add-ons are among the most advanced anywhere. Plug in the GIANT BOARD and you can record and play back programmes, edit and debug programmes, communicate with remote devices and make things happen in the outside world. Add Kluge Board to get ELF 11 to solve special problems such as operating a more complex alarm system or controlling a printing press. Add 4k RAM Board and you can write longer programmes, store more information and solve more sophisticated problems.

Expanded, ELF 11 is perfect for engineering, business, industrial, scientific and personal finance and Tax applications. No other small computer anywhere near ELF 11's low price is backed by such extensive research and development programmes.

The ELF-Bug Monitor is an extremely recent breakthrough that lets you debug programmes with lightning speed because the key to debugging is to know what's inside the registers of the microprocessor and, instead of single stepping through your programme, the ELF-BUG monitor, utilising break points, lets you display the entire contents of the registers on your T.V. screen at any point in your programme. You find out immediately what's going on and can make any necessary changes. Programming is further simplified by displaying 24 Bytes of RAM with full address, blinking curser, and auto scrolling. A must for serious programmers! Netronics will soon be introducing the ELF 11 colour graphics and music system — more breakthroughs that ELF 11 owners will be the first to enjoy!

## NOW BASIC MAKES PROGRAMMING ELF 11 EVEN EASIER!

Like all computers, ELF 11 understands only "machine language" — the language computers use to talk to each other. But, to make life easier for you we have developed an ELF 11 Tiny BASIC. It talks to ELF 11 in machine language for you, so you can programme ELF 11 with simple words that can be typed out on a keyboard such as PRINT RUN and LOAD.

## "ASK NOT WHAT YOUR COMPUTER CAN DO . . . BUT WHAT IT CAN DO FOR *YOU*"

Don't be trapped into buying a dinosaur simply because you can afford it and it's big. ELF 11 is more fun than "big name" computers that cost a lot more money.

With ELF 11 you learn to write and run your own programmes. You're never reduced to being a mere key punch operator working blindly with someone else's predeveloped software.

No matter what your speciality is, owning a computer which you really know how to use is sure to make you a leader. ELF 11 is the fastest way there is to get into computers. Order from the coupon below!

**H. L. Audio, 9b Garman Road, London N17 OUR**

SEND TODAY

NOW AVAILABLE FOR ELF 11

- ☐ Tom Pittman's Short Course on Microprocessor & Computer Programming teaches you just about everything there is to know about ELF 11 or any RCA 1802 computer. Written in non-technical language, it's a learning breakthrough for engineers and laymen alike. £5.00 post paid!
- ☐ Deluxe metal cabinet with plexiglasdust cover for ELF 11. £29.95* plus £1.50 p&p. R.F. modulator for use with T.V. set £3.00* post paid
- ☐ GIANT BOARD kit with cassette I/O, RS 232-C, TTY I/O, 8-bit P.I.O, decoders for 14 separate I/O instructions and a system monitor/editor. £39.95 plus £1.00 p&p
- ☐ KLUGE (prototype) BOARD accepts up to 36 IC's £17.00 plus 50p p&p
- ☐ 4k static RAM kit. Addressable to any 4k page to 64k £89.95 plus £1.56 p&p
- ☐ Gold plated 86-pin connectors (one required for each plug-in board) £5.70* post paid
- ☐ Professional ASC11 Keyboard kit with 128 ASC11 upper/lower case set, 96 printable characters, onboard regulator, parity, logic selection and choice of 4 handshaking signals to mate with almost any computer. £64.95* post paid
- ☐ Deluxe metal cabinet for ASC11 Keyboard £19.95* plus £1.50 p&p
- ☐ ELF 11 Tiny BASIC on cassette tape. Commands include SAVE, LOAD, ÷, x, +, (), 26 variables A-Z, LET, IF/THEN, INPUT, PRINT, GO TO, GO SUB, RETURN, END, REM, CLEAR, LIST, RUN, PLOT, PEEK, POKE. Comes fully documented and includes alphanumeric generator required to display alphanumeric characters directly on your T.V. screen without additional hardware. Also plays tick-tack-toe plus a drawing game that uses ELF11's hex keyboard as a joystick. 4k memory required £14.95* post paid
- ☐ Tom Pittman's Short course on Tiny BASIC for ELF11 £5.00* post paid
  Expansion Power Supply (required when adding 4k RAM) £19.95*
- ☐ ELF BUG TM Deluxe System Monitor on cassette tape. Allows displaying the contents of all registers on your T.V. at any point in your programme. Also displays 24 bytes of memory with full addresses, blinking cursor and auto scrolling. A must for the serious programmer! £14.95* postpaid

COMING SOON: A-D, D-A converter, Light Pen, Controller Board, Colour Graphics & Music System and more!

*Plus VAT at 8% on price and postage

CALL OR WRITE FOR WIRED PRICES

H. L. AUDIO LTD. DEPT E.T.1
9b GARMAN ROAD
LONDON N17 OUR
(TEL. 01-808 6127)
Sole European Distributors for Netronics
R&D Ltd. U.S.A.

YES! I want to run programmes at home and have enclosed ☐ £109.56 including postage and V.A.T. for RCA COSMAC ELF11 kit ☐ £5.94 including postage and V.A.T. for power supply (required) ☐ £5.95 including postage and V.A.T. for RCA 1802 Users Manual ☐ £5.95 including postage and V.A.T. for "Short Course On Microprocessor and Computer programming"

☐ I want mine wired and tested with power supply, RCA 1802 Users Manual and Short Course included for just £164.10 including postage and V.A.T. ☐ I am also enclosing payment (including postage and V.A.T.) for the items checked at the left.

Total enclosed

USE YOUR ☐ ACCESS ☐ BARCLAYCARD
Interbank
Account No

Signature . . . . . . . . . . . . . . . . Exp date

**CREDIT CARD PHONE ORDERS ACCEPTED 01-808 6127**

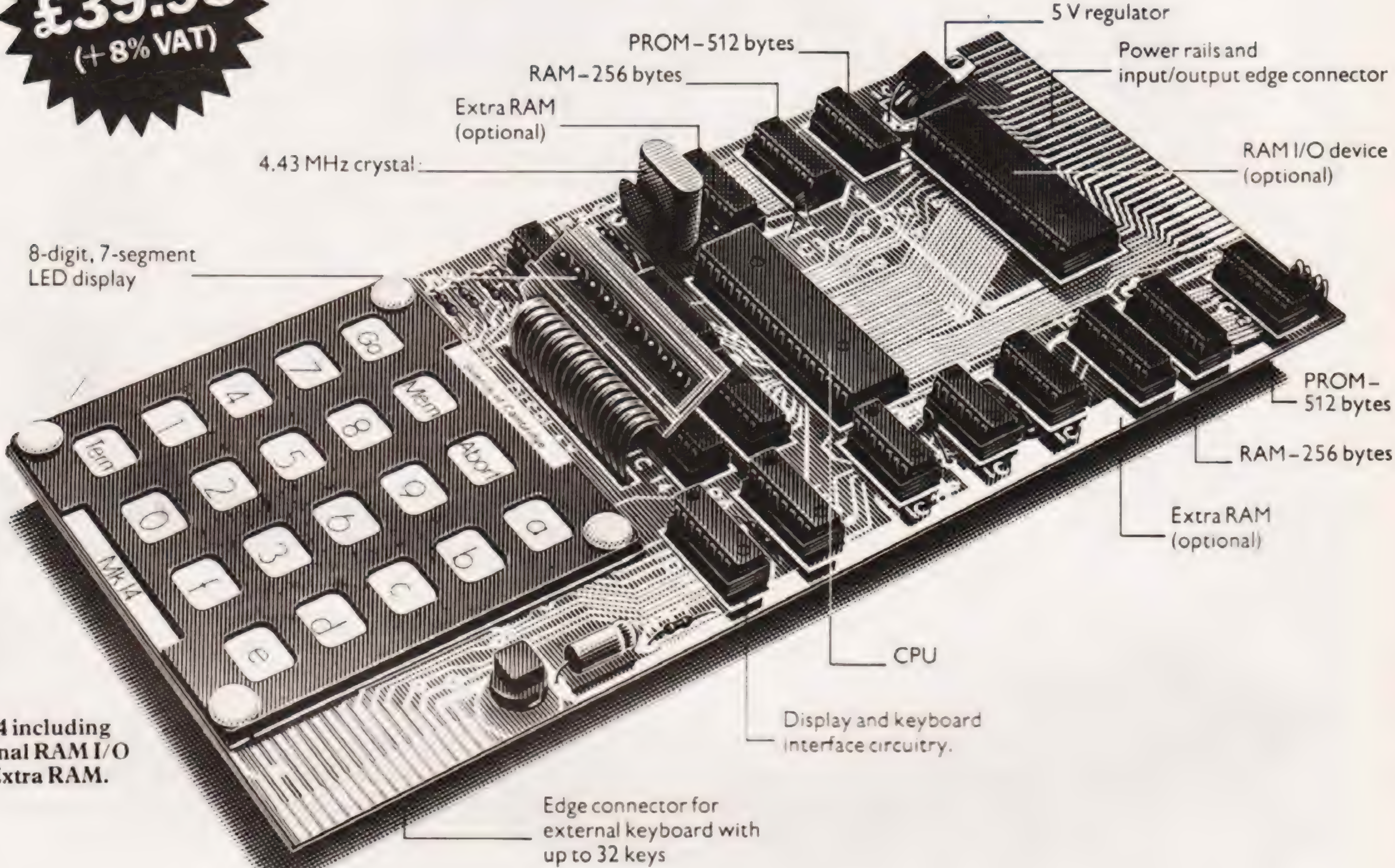
PRINT
Name

Address

DEALER ENQUIRIES INVITED

# From Science of Cambridge: the new MK 14.

**only £39.95 (+8% VAT)**




**MK 14 including optional RAM I/O and Extra RAM.**

### MK 14 – a complete computer for £39.95 (+8% VAT)

The MK 14 is a complete microcomputer with a keyboard, a display, 8 x 512-byte pre-programmed PROMs, and a 256-byte RAM programmable through the keyboard.

As such the MK 14 can handle dozens of user-written programs through the hexadecimal keyboard. (20 sample programs are provided in the Manual – which also contains comprehensive building instructions, and instructions on program-writing.)

Yet in kit form (which can be assembled by any fairly experienced kit-builder), the MK 14 costs only £39.95 (+£3.20 VAT, and p&p).

### But that's only the start....

The memory capacity of the basic kit is surprisingly powerful – but every computer owner, from a schoolboy to a multi-national corporation, soon feels the need for more memory.

With the MK14, it's yours!

Optional extras include an additional 256-byte RAM, and a 16-line external input/output device (allowed for on the PCB) which give a further 128 bytes of RAM.

### And the next step?

The next step is to add your own peripherals!

The first could be a low-cost module which provides an **interface with a standard cassette-recorder.** This means you can use ordinary tape-cassettes for the storage of data and programs.

To get the best from this configuration, you could uprate your system with a **revised monitor** – consisting of 2 replacement PROMs, pre-programmed with sub-routines for the interface, offset calculation and single step, and single-operation data entry.

The second peripheral could be your own **PROM programmer and blank PROMs** to set up your own pre-programmed dedicated applications. (Fusible-link device guarantees program safety.)

All are available now to owners of MK 14 – and remember Science of Cambridge keep you up to date *automatically* with advances in the MK 14 range. A TV interface device is already in the pipeline!

### A valuable tool – and a training aid

As a computer, it handles operations of all types – from complex games to digital alarm clock functioning, from basic maths to a pulse delay chain. Programs are in the Manual, together with instructions for creating your own genuinely valuable programs.

And, of course, it's a superb education and training aid – providing an ideal introduction to computer technology.

### SPECIFICATIONS

**MK 14**

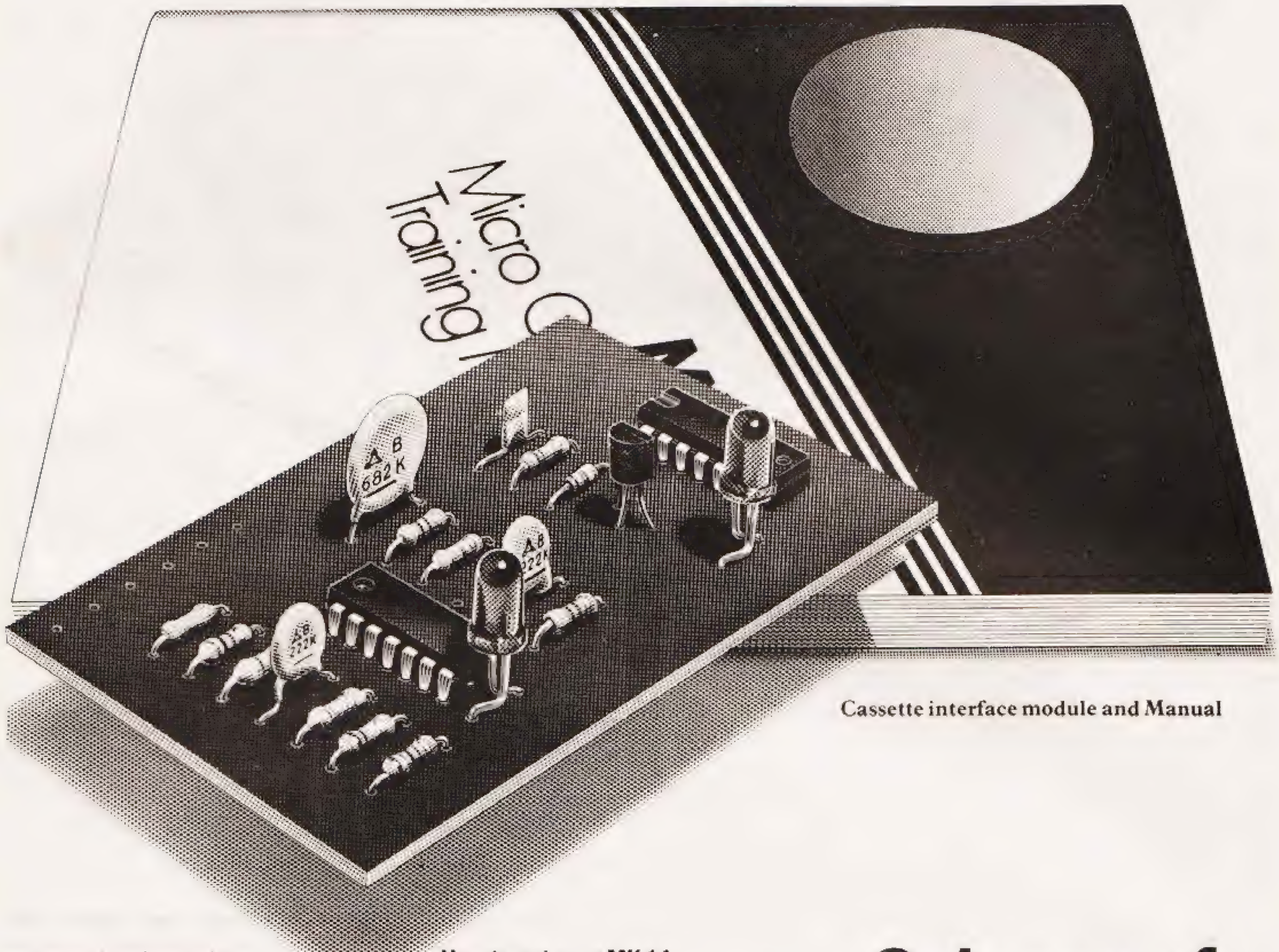
* Hexadecimal keyboard
* 8-digit, 7-segment LED display
* 8 x 512 PROM, containing monitor program and interface instructions
* 256 bytes of RAM
* 4 MHz crystal
* 5 V regulator
* Single 8 V power supply
* Space available for extra 256-byte RAM and 16 port I/O

Edge connector access to all data lines and I/O ports

**Optional Extras**

* Extra RAM – 256 bytes
* RAM I/O device

# Simplest, most advanced, most flexible microcomputer -in kit form.




Cassette interface module and Manual

* Cassette interface module
* Revised monitor
* PROM programmer
* Blank PROMS

**Free Manual**
Every MK 14 Microcomputer kit includes a Manual which deals with procedures from soldering techniques, through programming and use of RAM I/O to interfacing with complex external equipment. It contains operational instructions and examples for training applications, and numerous programs including math routines (square root, etc.), digital alarm clock, single-step, music box, mastermind and moon landing games, self-replication, general purpose sequencing, etc.

**Designed for fast, easy assembly**
Each 31-piece kit includes everything you need to make a full-scale working microprocessor, from 15 chips, a 4-part keyboard, display interface components, to PCB, switch and fixings.

The MK 14 can be assembled by anyone with a fine-tip soldering iron and a few hours' spare time, using the illustrated step-by-step instructions provided.

**How to get your MK 14**
Getting your MK 14 kit is easy. Just fill in the coupon below, and post it to us today, with a cheque or PO made payable to Science of Cambridge. And, of course, it comes to you with a comprehensive guarantee. If for any reason, you're not completely satisfied with your MK 14, return it to us within 14 days for a full cash refund.

# Science of Cambridge

Science of Cambridge Ltd,
6 Kings Parade, Cambridge, Cambs., CB2 1SN.
Telephone: Cambridge (0223) 311488

**To: Science of Cambridge Ltd, 6 Kings Parade, Cambridge, Cambs., CB2 1SN.**
Please send me the following, plus details of other peripherals:

☐ MK 14 Standard Microcomputer Kit @ £43.55 (inc 40p p&p.)
☐ Extra RAM @ £3.88 (inc p&p.)
☐ RAM I/O device @ £8.42 (inc p&p.)

I enclose cheque/money order/PO for £________ (indicate total amount.)

Name____________________

Address (please print)____________________

ETI 1

Allow 21 days for delivery.

# ETI BOOK SERVICE

## computing

**BASIC: A Self Teaching Guide (2nd Edition)** **£4.50**
by ALBRECHT, R. L.
Teach yourself the programming language BASIC. You will learn how to use the computer as a tool in home or office and you will need no special maths or science background

**Illustrating BASIC** **£2.20**
by ALCOCK, D.
This book presents a popular and widely available language called BASIC and explains how to write simple programs

**Microprocessors** **£10.65**
by ALTMAN, L.
Gives a general overview of the technology design ideas and explains practical applications.

**Applying Microprocessors** **£12.00**
by ALTMAN, L.
Follow up volume which takes you into the second and third generation devices

**Intro to Microprocessors** **£6.00**
by ASPINALL, D.
Explains the characteristics of the component

**How to Buy and Use Minicomputers and Microcomputers** **£7.50**
by BARDEN, W.
Discusses these smaller computers and shows how they can be used in a variety of practical and recreational tasks in the home or business

**How to Program Microcomputers** **£6.75**
by BARDEN, W.
This book explains assembly-language programming of microcomputers based on the Intel 8080, Motorola MC6800, and MOS Technology MCS6502 microprocessors.

**Introduction to Microcomputers and Microprocessors** **£7.50**
by BARNA, A.
Provides the basic knowledge required to understand microprocessor systems. Presents a fundamental discussion of many topics in both hardware and software

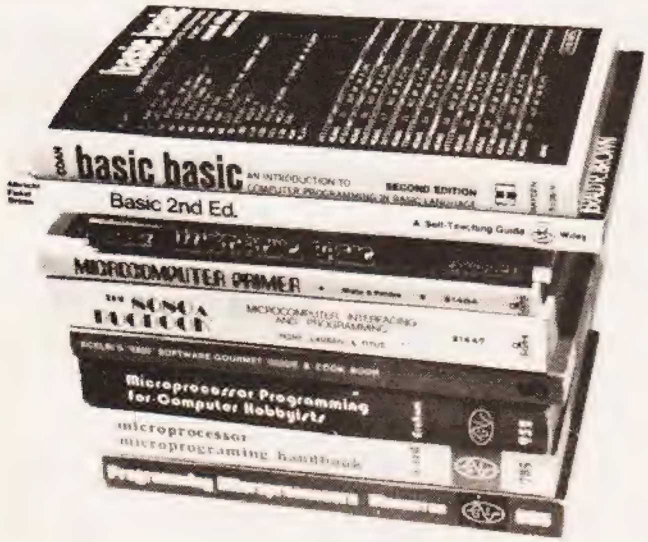
**Microprocessors in Instruments and Control** **£11.80**
by BIBBERO, R. J.
Introduces the background elements, paying particular regard to the dynamics and computational instrumentation required to accomplish real-time data processing tasks

**Basic BASIC** **£7.50**
by COAN, J. S.
An introduction to computer programming in BASIC language

**Microprocessor Programming for Computer Hobbyists** **£7.00**
by GRAHAM, N.

**The Computer Book** **£6.20**
by HAVILAND, R. P.
Building super calculators and minicomputer hardware with calculator chips

**Microcomputers, Microcomproesors, Hardware Software and Applications** **£14.25**
by HILBURN, J. L.
Complete and practical introduction to the design, programming, operation, uses, and maintenance of modern microprocessors, their integrated circuits and other components




**Microprocessor Systems Design** **£14.35**
by KLINGMAN, E.
Outstanding for its information on *real* microprocessors, this text is both an introduction and a detailed information source treating over a dozen processors, including new third generation devices. No prior knowledge of microprocessors or microelectronics is required of the reader.

**BASIC Programming** **£6.10**
by KEMENY, J. G.
A basic text.

**Microprocessor and Small Digital Computer Systems for Engineers and Scientists** **£19.00**
by KORN, G. A.
This book covers the types, languages, design, software and applications of microprocessors.

**TV Typewriter Cookbook** **£7.40**
by LANCASTER, D.
An in-depth coverage of tv typewriters (tvt's)—the only truly low-cost microcomputer and small-system display interface. Covers tvt terminilogy, principles of operation, tv configurations, memories, system design, cursor and update circuitry and techniques, hard copy, color graphics, and keyboards and encoders

**Microprocessors — Technology, Architecture, and Applications** **£8.00**
by McGLYNN, D. R.
This introduction to the "computer-on-a-chip" provides a clear explanation of this important new device. It describes the computer elements and electronic semiconductor technologies that characterize microprocessors.

**Programming Microprocessors** **£5.50**
by McMURRAN
A practical programming guide that includes architecture, arithmetic logic operations fixed and floating-point computations, data exchange with peripheral devices, compilers and other programming aids

**Microcomputer Based Design** **£19.00**
by PEATMAN, J. B.
This book is intended for undergraduate courses on microprocessors.

**Microprocessor and Microprocessor Systems** **£20.50**
by RAO, G. L.
A completely up-to-date report on the state of the art of microprocessors and microcomputers, written by one of the leading experts. It thoroughly analyzes currently available equipment, including associated large scale integration hardware and firmware

**The 8080A Bugbook: Microcomputer Interfacing and Programming** **£7.60**
by RONY, P. H.
The principles, concepts and applications of an 8-bit microcomputer based on the 8080 microprocessor IC chip. The emphasis is on the computer as a controller.

**6800 Software Gourmet Guide and Cookbook** **£7.80**
by SCELBI

**8080 Software Gourmet Guide and Cookbook** **£7.80**
by SCELBI

**Understanding Microcomputers** **£7.60**
Gives the fundamental concepts of virtually all microcomputers
by SCELBI

**Microprocessors and Microcomputers** **£18.00**
by SOUCEK, B.
Here's a description of the application, programming, and interacing techniques common to all microprocessors. It concentrates on detailed descriptions of representative microprocessor families and includes explanations of digital codes, logical systems, and microcomputer organization

**Microcomputer Primer** **£6.25**
by WAITE, M.
Introduces the beginner to the basic principles of the microcomputers. Discusses the five main parts of a computer — central processing unit, memory, input output interfaces, and programs. The important characteristics of several well-known microprocessors are given and a chapter is included on programming your own microcomputer

**Microprocessor/Microprogramming Handbook** **£6.00**
by WARD
Authoritative practical guide to microprocessor construction, programming and applications

**HOW TO ORDER:**
**Make cheques etc payable to ETI Book Service. Payment in sterling only please. Orders should be sent to: ETI Book Service, PO Box 79, Maidenhead, Berks. All prices include P & P.**

# Softspot

**Don Scales has written another game for the TRITON. We would like to expand Softspot over the next few months so please send us any software (machine code, BASIC etc.) which you feel would be of interest to others.**

REVERSAL (1K)
DESCRIPTION

The REVERSAL GAME is a game of skill played with the computer. The computer will arrange the numbers 1 to 9 in a random sequence and your job is to arrange them into ascending order in the minimum number of moves.

For example

The machine might print the following sequence –
8 3 7 9 1 4 2 6 5

You now have to get these arranged in ascending order. To do this, you are allowed to reverse the order of any number of digits starting at the left hand side. The machine will ask

NUMBER TO REVERSE

If you enter 4 the result will be

9 7 3 8 1 4 2 6 5

If you now enter 9 you will get

5 6 2 4 1 8 3 7 9

You now have 9 in the correct place and can set about getting 8 7 6 etc. in their correct positions.

```
10 PRINT 'THE REVERSAL GAME'
20 FOR I=1 TO 9
30 LET A=RND (9)
40 IF I=1 GOTO 80
50 FOR J = 1 TO I - 1
60 IF @ (J) = A GOTO 30
70 NEXT J
80 LET @ (I) = A
90 NEXT I
95 LET B=0
200 PRINT #2, @ (1), @ (2), @ (3), @ (4), @ (5), @
(6), @ (7), @ (8), @ (9)
210 INPUT 'NUMBER TO REVERSE' J
220 IF J <1 GOTO 240
230 IF J <10 GOTO 260
240 PRINT 'INVALID – TRY AGAIN'
250 GOTO 210
260 LET K=(J+1)/2
270 FOR I=1 TO K
280 LET A = @ (I), @ (I) = @ (J+1-I), @ (J+1–I)
= A
290 NEXT I
295 LET B = B+1
300 FOR I=1 TO 9
310 IF @ (I) # I GOTO 200
320 NEXT I
330 PRINT 'TOTAL', #3,B
340 GOTO 10
```

# Personal Computing - The Early Years

**Tip toe through the early memories of the Personal Computing field**

## In The Beginning

It all started back in 1974 when Intel introduced the 4004, the first true microprocessor. It developed almost by accident, as a result of Intel's efforts to produce a calculator of unprecedented flexibility. The shock waves of the hand-held calculator revolution were still being felt by every section of society and the back-room boys' eyes lit up with that "You ain't seen nothing yet" look as they drew up the chip-masks for their next product, the 8-bit 8008.

Assisted by the lessons learned from the cut-throat calculator business the microprocessor developed with frightening speed and predictability. Frightening not only because of the vast amount of high-technology and high finance poured into the field, but also because of the dramatic effect extrapolations of such technology can have on a broad spectrum of society. Predictable because everyone knew what was going to develop. The shrinking of calculators from giant cabinets to flip-top packs in the space of just a few years created an extraordinary blase attitude towards electronics. It was a spectacular demonstration of the omnipotence of the new technology of micro-electronics. It was proof that now nothing is impossible — just draw up the specifications, expend x thousand man-hours and y million dollars, and there you have it. So the microprocessor has been born at a time when nothing comes as a surprise any more. But maybe the real surprise is yet to come.

## The New "Hams"

Of all the sub-sets of electronics hobbyists the most clearly defined to date has been the radio amateur. Members of this particular sect follow a technological faith which started with the first wireless communication and has since flourished, gaining millions of followers in a relatively short time.

For many hams their hobby almost becomes a life-style within itself, always striving for that rare DX the eternal pursuit of that elusive one-to-one SWR. Is it possible that we are now witnessing the founding of a new faith, one whose god speaks in 1s and 0s rather than 5s and 9s?

By making a few comparisons between amateur radio and amateur computing certain patterns can be seen emerging which may be an indication of what course the future of personal computing might take.

## Power To The People

Like amateur radio, amateur computing is a high-technology which makes the latest developments in the science of electronics available to anyone at all who has the time and money to pursue them. The money factor is all important — the lower the cost of the hardware, the more people can afford to pursue the hobby. A reasonably useful microcomputer system might cost in the order of $1,000. Hams might spend this sort of money on radio gear, for that matter a radio-controlled aircraft enthusiast, amateur photographer or stamp-collector could easily spend that much on his hobby. So thanks to the microprocessor the cost of your own personal computer is no longer a barrier to most people.

This new accessibility and the free interchange of ideas and information between hobbyists has the effect of distributing "computer power" over a broad spectrum of people. This leads to a breaking down of much of the mystique which has traditionally surrounded the world of computers; they are being de-mystified as the magic is systematically exposed as little more than sleight of hand. As large number of amateurs invade a hitherto sacred field which was once the sole province of a privileged few the elite will inevitably grow in number until it finally becomes plebian.

## "Homebrew" vs "Appliance"

As with amateur radio there are two factions within the computer cult, the "homebrewer" who builds his own equipment for the sake of the experience gained, and the "appliance operator" who buys a ready-built, going unit and gets what he wants from operating his instrument, writing programs and experimenting with the performance of the hardware as bought. His investment in the computer itself is more financial and less emotional than in the case of the homebrewer.

There is always some overlap between the two factions, but they can usually be classified by comparing the time spent building, testing and modifying the hardware to the time spent actually using it once it's working.

## The Sky's The Limit

Radio equipment has rather unique and interesting characteristics. It can never really be declared "finished". There is always more to add to the station, improvements to be made, better antennas, higher power, lower noise. Computers share this trait which makes them too the ideal subject for a hobby. Today's mass storage is tomorrow's scratch-pad. There is unlimited scope for improvement and expansion of the hardware.

If ever the computer itself should look like having its full complement of RAM, ROM. AND I/O parts, the hobbyist can turn his attention to the vast range of peripherals that are available to him. A radio transmitter can be hooked up to an antenna and a microphone and that's about it, but nothing can be so insular as to resist interfacing to a computer if the intrepid hobbyist uses a little imagination.

More importantly, once the computer is operative a literally infinite amount of software development waits to be done. Like radio operating, this phase of the hobby is particularly attractive because the operating cost is nothing more than the electricity bill.

## The Junk Box

Ever since the tradition of stripping a discarded radio chassis was established by the pioneers of amateur radio, the humble junk box has been the hallmark of the truly worthy hobbyist. In much the same way as one may judge someone's social standing by the way he dresses, how neatly his garden is kept, radio amateurs assess each other's status by the quality and quantity of a bits and pieces which lurk for years in the dark recesses of their junk box until their true worth is finally recognised and they are discarded. Because the microcomputer hobby is so new, junk box computer parts of good vintage are rarer, but there is always the stimulating challenge of pushing a seemingly irredeemable piece of obsolete equipment into service. Radio ham and computer hobbyist alike share the unique pride and joy of operating equipment which the professionals have officially declared worthless.

## Doing The Impossible

Besides the resurrection of dead equipment, hams are keen on performing another type of miracle. This involves proving by practical demonstration that something which should by rights not be possible does, in fact, work. With amateur radio this usually entails forging forth into extremes of technology (or bad practice, depending on how you look at it), generally revolving around a successful communication in spite of a red-hot "final", vast distances or an antenna made of wet string.

To the computer ham comparable feats entail successful execution of programs which are either exceptionally short or unbelievably long or so cunningly convoluted that not even the person who wrote it knows how it works. Thanks to the new technology involved there is also a whole new set of miracles which rely on getting a phenomenal number of logic functions into an incredible small space.

Amateurs are in a rather unique position in that they are permitted to exceed manufacturers' ratings to see to what limits they can push a particular component or piece of equipment. This practice gave rise to many novel techniques in the field of radio and a similar thing is bound to happen in computing.

**The software bugs seem to come out just before sunrise.**

## Time Is Not Money

Amateurs make many other contributions to the science to which they are devoted as a result of the enormous amount of time they spend on their hobby. Because of the non-commercial nature of their pursuits, computer hobbyists can afford to undertake time-consuming projects which would not be economical as a professional enterprise.

Like the radio amateur who stays awake all night tuning across the bands looking for a rare contact, the computer ham often burns the midnight oil chasing an elusive bug in his software. Radio propagation never seems to be optimum at a civilized hour; similarly the software bugs only seem to come out just before sunrise.

With both amateur radio and amateur computing the real fun of the hobby lies in setting a goal and then achieving it no matter how long it takes or how inefficient the techniques used may be. The computer ham may devote hundreds of hours to developing a program that does nothing more than play a seemingly useless game. But, as with any technical hobby, a lot of valuable techniques are learned in the process.

## Spreading The Word

A natural development from any widely followed hobby is the formation of clubs where people with similar interests can meet and exchange ideas. Major amateur radio clubs like the Radio Society of Great Britain, the Amateur Radio Relay League and the Wireless Institute of Australia have been established for many years and cater for hundreds of thousands of enthusiasts.

Even though the do-it-yourself computer hobby is

# – The Early Years

so young there are already hundreds of computer hobby clubs. The biggest of these are found on the west coast of America which is where most of the world's microprocessor products originate. The Southern California Computing Society has about 5,000 members. At the moment there are nearly 200 smaller computer clubs in the USA and an estimated 20,000 people have their own personal computer.

Magazines devoted entirely to the computer hobbyist have been established with great success. The most widely read glossy is **byte** which now circulates over 60,000 copies.

The radio amateurs' "field day" has always provided a means of information exchange between individuals. As communication is the basis of ham radio, publicising such events poses no problems, but computer hams have only their specialist magazines for such promotion. A few conventions have been held by computer hobbyists where the main purpose has been to establish standards so that hobbyists can easily share the software they have developed. Manufacturers of personal computing hardware also take an interest in these gatherings because it is an excellent opportunity to find out what the hobbyist is interested in and therefore which products will sell.

## Speaking of Computers

Due to the unusually verbal nature of the hobby itself, radio amateurs have developed a unique vocabulary. The language which results has such a high jargon content and is spoken so fluently that it is quite unintelligible to the outsider. This serves to give the group its own identity and binds its members together.

Although amateur computing is still in its infancy its followers found that the computer industry had already provided them with a highly developed jargon, complete with an impressive range of off-the-shelf, buzzwords which have been nutured to perfection by 20 years of professional verbal dazzling. This they have eagerly seized and followers now have a language of their own.

The most telling sign of both radio and computer hams is their often amusing ability to construct seemingly meaningful sentences using all the rules of English grammar except that the keywords are replaced with strings of number of initials. The radio amateur might say, "QRX, I've got to check my SWR", while the computer amateur could hit you with, "I've put a PIA on my 6800 for I/O."

To the uninitiated talking in code like this seems like an awfully anti-social way of passing secret messages between club members — it serves to keep the in-group "in" by providing a feeling of comradeship for members and it keeps ot all but the most determined newcomers.

## Future Shock (Electric)

Although personal computing is already well established as a hobby, the real impact of its advent is yet to come.

It is a characteristic of any hobby that those who pursue it develop great expertise in the field.A keen 10 year old stamp collector may know as much about stamps as a professional stamp dealer. Having spent his youth building radio transmitters a ham of 20 might know as much about radio as a University-qualified electronic engineer.

We are now finding a new breed of hobbyist/expert, a hobbyist who has spent thousands of hours of leisure time building computers and programming them. He could well know more about computers than many professionals in the field. As the hobby grows there will be more and more people to whom computers are second nature, people who are fully conversant with a broad range of computer concepts and totally up-to-date with the state of the art.

Traditional training and qualifications are already being seriously challenged by these hobbyists who might enroll in a University computer science course already knowing more than they will be taught.

As this flood of expertise hits the workforce we are bound to see dramatic changes in the status of the computer professional. Will there be a sudden surplus of computer engineers and programmers, or will the wave of new technology bring with it expansion of the industry to absorb it?

The remarkable advances in solid state technology which led to the development of microprocessors have made their mark on the electronics industry, but it's the "expertise explosion" which will follow that will have the real impact on society.

There are rumours of an Opto-Pneumatic Power Cue being evaluated by leading players. Integral Laser Theodolite system determines range and angles while Microprocessor backpack computes and controls shot dynamics.

# TRS–80 Update

The August issue of ETI carried a reciew of the Tandy's TRS-80. The machine, made available for review, was equipped with the Tandy Level 1 floating point 'Tiny BASIC' ROM and 4K of dynamic RAM.

Since then Tandy have introduced a number of developments on the BASIC TRS-80 machine — The Level 1 system was made available with 16K of memory and the more powerful Tandy Level 2 BASIC was introduced into this country supported by both 4K and, top of the range, 16K of dynamic RAM. Any machine from the range can be upgraded in performance by returning it to Tandy for internal mods.

This update to the original review deals with the major differences between machines equipped with the Level 1 and Level 2 BASICs. For a general description of the TRS-80 machine, the comments of the original review are still valid.

## Key Improvement

The level 2 BASIC is a 12K version of the microsoft BASIC used by many machines in the TRS-80's price bracket. Before going on to describe the Level 2 BASIC, which is considerably more powerful than Level 1, in detail, it may be as well to mention two hardware orientated improvements implemented on this upgraded machine.

The TRS-80's keyboard is decoded by the resident software and in the Level 1 machine the decoding system used meant that one had to release one key before the computer would allow the entry of another. This led to such messages as "Plase tye in yor name" — Even those of us who are not touch typists could get up enough speed to defeat the Level 1 keyboard. Level 2 lets you hit the second key before you have released the first key. Even level 2 does not let you get away with too much, however, and if you hold down three or four keys at a time, some keys will then generate several characters as they are pressed.

The other major area of improvement on the hardware/software border is the speed at which programs are dumped to and loaded from the TRS-80's cassette recorder.

Level 1 used a data transfer rate of 250 baud — the level 2 rate is twice this at 500 baud. Because of this faster transfer rate it may be found that the volume setting on the recorder that was suitable for use with a Level 1 machine will have to be altered for satisfactory performance with Level 2, in general a lower volume control setting will be needed.

## Editing

One of the major differences between Level 1 and Level 2 is the provision of a powerful editing facility in the latter. We do not have enough space to go into a detailed description of the editor but a few examples of what can be accomplished should give an idea of its power.

One can list any line of a program individually, insert material anywhere in a line, delete material anywhere in a line, delete the remainder of a line beyond the cursor and insert new material in its place. Any desired number of characters to the right of the cursor may be deleted or a change may be made to a specified number of characters. The editor provides a search facility, so that (for example) a line may be searched for the second occurance of the letter G and move the cursor to that position or tell the computer to delete all characters to that point and leave the cursor there.

Other editing features allow you to quit the edit saving all changes, to quit deleting all changes or the cancel all changes made and restart.

All in all a very powerful editor.

Level two basic allows variables to be stored in different forms — single or double precision. A # (hash) after a calculation will cause the result to be output as a 16 place decimal (print 1/3 #), an ! (exclamation mark) will keep a variable at single precision (G!) and a % (percent) symbol will keep the number as an interger.

The system of representation can be selected for the type of work required — Tip, if you can work with whole numbers, store them as integers, your program will run twice as fast and occupy half the memory space as programs without these restrictions.

## Print Format

Level 2 BASIC allows some fairly sophisticated formatting of output. It can be used in many applications such as printing report headings, accounting reports etc. Using nine "Field Seperators," one can specify digit positions, cause automatic rounding off, concatentate (join together) multiple strings or string variables, align columns — a comprehensive system that allows any output to be presented in an easily readable form.

## Strings

The Level 2 manual states that "Without string handling capabilities, a computer is just a superpowered calculator." What this makes the level 1 machine with it's "String Things" — strings which one was not able to handle — is a question we leave open: Needless to say the Level 2 machine provides a wide range of string handling commands.

The DEFSTR statement allows any valid variable name to contain a string, adding a type declaration character ($) has the same effect.

Each string can contain up to 255 characters and strings can be compared as well as concatented.

Strings can be compared with the same symbols used for comparing numbers — The ASCII codes for the characters being the values compared.

Level 2 features ASC and CHR$ commands. ASC gives the ASCII numerical code for a string character while CHR$ performs the reverse operation.

The INKEYS$ function will allow an entry from the keyboard while a program is running — without the use of the return key, useful for "Real Time" games if nothing else. INKEY$ will strobe the keyboard and return with a one character string — This being a null string if no key is pressed.

Manipulation of strings can be carried out with the following commands, LEFT$, MID$, RIGHT$, LEN and string$.

LEFT$ (A$, 3) will print the first three characters from the left of A$ — Thus if A$ were TANDY the command would select TAN — MID$ will select characters from the middle of the string and RIGHT$, not surprisingly, from the right. The formats are the same for LEFT$.

STRING$ provides a string of a specified character for a specified charact4er for a specified length. For example STRING$ (25, ?) would output 25 question marks.

## Trace

Level 2 BASIC provides a trace facility that is very useful for the debugging of programs. The command TRON followed by RUN will output to the screen the exact sequence in which the program lines are executed. To turn off the trace function the command TROFF is used.

## Functions

Level 2 adds considerably to the four in- built arithmetic features of level 1.

To MEM (size of used memory), INT (Convert variable to interger), ABS (absolute value of variable) and RND (Random number generator) are added 12 more.

These are the trig functions SIN, COS, TAN and ATAN — The maths functions EXP, LOG, SGN and SQR.

The command RANDOM at the beginning of a program will ensure a different series of random numbers each time program is run. CDBL and CSNG concern themselves with the format in which a numerical variable is stored. CDBL provides a double-precision value of the expression following CDBL in brackets, even if the operands are single — precision or integers, CSNG does the oposite by providing a single precision value of the expression.

Many other functions may be created using the 16 Level 2 functions and Appendix E of the level two manual provides a guide to these.

## Error messages

Whereas level 1 BASIC provides three (WHAT? HOW: and SORRY) error messages, Level 2 has 23 two-letter codes providing a far more specific indication of the error. Level l's feature of printing the error code at the exact point at which the error occurs is however lost — The message being printed on the following line.

Another facility present in Level 1 but lost in the more powerful version is that of abbreviated statements and commands.

Tandy, however, produce a conversion-tape which will allow software written for a Level 1 machine to be converted to run a Level 2 machine.

## Arrays

Arrays are permitted to Level 2, the number of dimensions being limited by the size of the available memory-string arrays are also allowed.

# TRS–80 Update

Level 2 BASIC provides the PEEK and POKE commands. Poke allows a specified value to be written into a specific RAM location PEEK allows the value stored at any RAM location to be retrieved.

POKE is particularly useful when producing graphics displays on the screen. Level 1's SET statements for handling screen graphics were rather slow — Level 2 provides 63 special graphics characters that can be speedily manipulated by the POKE command.

The display can also produce double width (same height) characters. By hitting shift key and right arrow the format is changed. Note however that anything on screen at the time only has every other letter enlarged. The clear key will return the display to the normal format.

## User Subroutines

Level 2 features the BASIC USR statement that permits the calling of user written subroutines.

## File Search

Level 2 allows the user to label files and to search for a the named file. CLOAD "TEST" will ignore all files on tape until the one called "TEST" is found. A useful feature is that as the machine searches for a file, the names of all those on the tape before the specific file is found will be displayed in the upper right hand corner of the display.

A file can be verified after being dumped by entering CLOAD? The machine will then load the program from tape and compare it with that stored in memory. A handy feature that allows one to ensure a program has been faithfully recorded.

## Manual

Unlike the level 1 manual, which makes an excellent job of teaching BASIC, the level 2 manual — As the forward says — Is not for the total beginner.

The manual does not go into the detail of the 233 page Level 1 manual. The Level 2 manual for example presents nine subroutines for array/matrix manipulation with very little explanation. 31 function codes are mentioned in one of appendices with little indication of how they are used in a program.

## Computer Conclusion

The Level 2 package certainly provides a significant improvement over the Level 1 version of this machine and, in our opinion takes the TRS-80 from the realms of the "Superpowered Calculator" — Tandy's words — into the area of real computing.

The number of add-ons available now — floppies, memory, printers etc means that the TRS-80 can form the heart of a flexible system suitable for a wide range of applications.

**See the article on a small business application of a TRS-80 system elsewhere in this issue.**

# THE MPU MAN

**KEY**

1 Teletype unit.
2 Punched paper tape reader.
3 Floppy disc 'player' unit.
4 Home-made floppy disc experimental unit.
5 Incompatible systems interface patch-board.
6 Skeleton keyboard for manual entry.
7 Modulator for feeding display material to domestic magic lantern.
8 Domestic magic lantern showing results of simulated World cup final.
9 Another advanced game in progress.
10 Working microprocessor module.
11 Well ventilated reprogrammable read only memory.
12 Interesting teleprinter graphics — genuine system check — no kid!
13 Early home-made 8-bit processor.
14 Power supply with outboard cooling and attractive economy housing.
15 Another development module. Seems to have undergone pre-frontal lobotomy. . . . Last Night of the Proms?
16 Junked cassette player with sprinkling of cassette programmes. Alternative to papertape or manual entry.
17 Primitive one-shot switch input unit.
18 Aerosol spray for de-bugging.
19 Mighty de-bugged noughts and crosses programme listing being triumphantly brandished.
20 Attempt to fabricate home brew floppy disc.
21 Another teletype unit.
22 Personalised low density punched tape storage system.

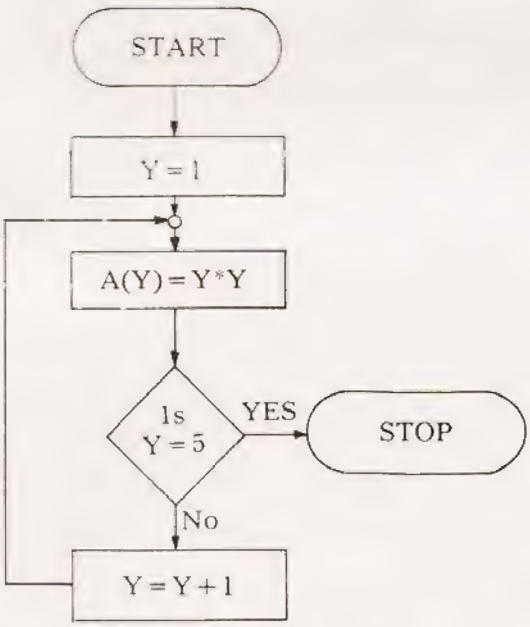
# Beginning BASIC

**Phil Cornes resumes his description of BASIC with a look at some of the conditional branching instructions featured in this language**

## IF THEN

THIS IS THE first of BASIC's really powerful conditional branching statements (we look at the others below) that go into the make-up of BASIC and we will add an IF THEN statement to the previous program segments (last month) to see what it can do.

Consider the following—




This is the same flow chart that we saw earlier except that now there is a two-way branch added which is made dependent upon the answer to the question 'IS Y = 5'.

Before we go on to look at the program derived from this flow chart, there is one other thing we need to consider. You will notice from the flowchart that IF Y is 5 when the decision box asks the question THEN we branch to a stop box. The statement in BASIC which causes the execution of a program to terminate is the END statement and you will find one of these in the program.

There is no statement in BASIC which corresponds to the start box on the flow chart (that is just provided for our information) and so the first box we consider contains Y = 1. The statement needed to convey this to the computer is—

LET Y = 1

but remember that every statement in a program must have a line number, and so we have—

10 LET Y = 1

We now move on to the next box

A(Y) = Y*Y and produce the statement—

20 LET A(Y) = Y*Y

The third box is the new one and we write—

30 IF Y = 5 THEN

THEN what? Well, we have to branch to the line number which contains the END statement, but we don't yet know which one this will be. So we can either sit and wait until we have written the END statement, or we can say always let the END statement exist on some high numbered line (say 9999) so if we ever need an END statement, we know what line it will appear on. We will do it this way so line 30 will read—

30 IF Y = 5 THEN 9999

and so if Y does equal 5, then we branch to the END statement that we will put in line 9999.

If the test (IS Y = 5) fails (answer is NO) then line 30 will be ignored and the computer will carry on executing the statements in the normal line number order.

The next box down contains Y = Y + 1, and so line 40 reads—

40 LET Y = Y + 1

From this we now branch back to the statement A(Y) = Y*Y which is on line 20 and we get—

50 GOTO 20

and lastly

9999 END

If we write this out in line number order, we get—

```
10 LET Y=1
20 LET A(Y)=Y*Y
30 IF Y=5 THEN 9999
40 Y=Y+1
50 GOTO 20
9999 END
```

and this is our first complete program.

It does not matter that the line numbers do not follow on in multiples of 10, they don't have to, but what is more important is the fact that we do leave some numbers spare between our statements so that if we find we have missed out a line, or think of something else that we would like to add, then we have plenty of space to do so.

Consider the following—

```
 5 REM INITIALIZE Y
10 LET Y=1
15 REM PUT Y*Y IN A(Y)
20 LET A(Y)=Y*Y
25 REM TEST FOR Y=5
30 IF Y=5 THEN 9999
40 LET Y=Y+1
50 GOTO 20
9999 END
```

REM (I thought REM was an android or something to do with sleep) in BASIC is short for REMark and tells the computer that whatever follows on this line is to be ignored because they are only notes for the programmer as a reminder of what is happening.

REMark statements in a program of this length are unnecessary, but we will soon be writing programs of sufficient length and complexity to justify their use as memory aids.

Returning now to our IF-THEN statement (IF Y=5THEN 9999) the equals sign used here is not an arithmetic operator, but the first of the comparison operators. Any of the other comparison operators (<,>,<=,>=,<>) could also be used in an IF THEN statement, so that—

```
30 IF Y>4 THEN 9999
130 IF Q<19.2 THEN 55
902 IF A(17)>=14.9 THEN LET P=P+1
```

are all valid statements.

Notice here the twist in the tail of line 902. This is also a valid statement on most machines. This is easier to understand if we consider the IF-THEN statement as two separate statements. The first part (the IF part) asks a question (in line 902 – IS A(17)>=14.9) to which the computer can answer either YES or NO. If the answer is NO then this statement is finished with and control passes on to the next higher numbered line. If the answer is YES then the computer passes on to the second statement on the line, the THEN part. THEN what? THEN LET P=P+1 or THEN END or THEN 900 (this is really an abbreviation of THEN GOTO 900) or THEN any other statement. We can even put another IF THEN statement in.

Consider the following—

```
200 IF (A=1)*(B=1) THEN 900
```

The computer encountering this would first ask the question IS A=1. If the answer is NO control passes to the next higher numbered line. If, on the other hand, A is equal to 1, we move on to the statement following the THEN and encounter another IF THEN statement which is treated in exactly the same way as the first. IS B=1. If NO then carry on with the next line, if YES THEN GOTO 900. You will see that using this logic we will only pass control to line 900 if both A=1 AND B=1. At about this point your memory should be stirring to the fact that you have read something about logical operators earlier and indeed this is the place where they fit in. Depending on which machine you are considering, there are two ways of re-writing line 200 above to achieve the same result.

You could use—

```
200 IF A=1 AND B=1 THEN 900
```

which will normally be the format for machines with standard or extended BASIC, or—

```
200 IF (A=11)*(B=1) THEN 900
```

for the tiny BASIC machines

Notice the brackets in the second example. These tell the computer where one comparison ends and the other starts, otherwise the computer would attempt the following—

```
200 IF A=1*B
```

(multiplication sign!) and then bomb out on the second equals sign.

The other common logical operator (OR) can also be used in a similar manner—

```
300 IF Q>3*H OR S<9 THEN R=R–2
```

or

```
300 IF (Q>3*H)+(S<9) THEN R=R–2
```

Notice the brackets again in the second example for similar reasons, and notice also the omission of the optional LET keyword before the R=R–2. We will continue to omit the LET from now on.

Finally for this month, we will go on to consider one of the pre-defined functions of BASIC (somewhat out of turn, but we'll see why in a moment) the random number generator.

## RND(X)

We will start off by saying that the X within the brackets (the brackets are necessary and must be used whatever we replace X by) may be replaced by

# BASIC

any constant, variable name or expression with the proviso that when the computer evaluates the contents of the brackets they must not be negative or the computer will bomb out. When the computer has evaluated the brackets and checked that the answer is not negative, anything after the decimal point is chopped off (so that 0.238 would be truncated to 0, similarly 8.9 would become 8 and so on) if the result of this operation is zero, then the computer will generate a random decimal number (up to 6 digits) in the range zero to one, so that when—

```
20 R=RND(0)
```

is executed, R will take a random value between zero and one. If the result after truncating the contents of the brackers is not zero (it must by now be a positive integer) then the computer will generate a random integer between one and the number in the brackets inclusive, so that, for example—

```
50 R=RND(6.8)
```

Would assign a random integer to R with a value between 1 and 6 inclusive (6.8 would be truncated to 6). This is a very useful function for any statistical or games applications and has been included at this time so that we can set you some homework (you need the practice). You will find that you now know enough about BASIC to convert the three card shuffling routines presented in the first part into programs and we would suggest that if you are following this series seriously, you should attempt to do just this. Sample answers will be presented next month.

The answers to the questions posed last month are

1 The expression has a value of 21, and
2 the expression could be simplified to
  7+7*8/2/((12+8)*2/20)
  You cannot remove the brackets round (12+8)*2/20 (if you made this mistake, think about why not).

Next month we go on to look at how we get the computer to print some answers, subroutines and some more conditional branching.

**CTI**

# AD INDEX

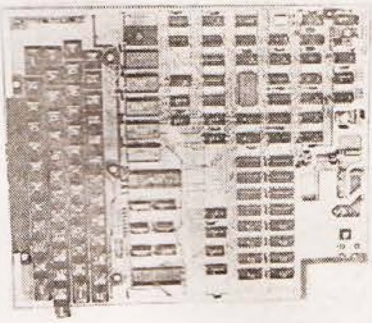
# The Age of Affordable Personal Computing Has Finally Arrived

Ohio Scientifics

## Superboard II

**Full 8K basic and 4K user RAM**
**Built and tested**

**£284.95**
inc. VAT & P&P



Ohio Scientific has made a major breakthrough in small computer technology which dramatically reduces the cost of personal computers. By use of custom LSI micro circuits, we have managed to put a complete ultra high performance computer and all necessary interfaces, including the keyboard and power supply, on a single printed circuit board. This new computer actually has more features and higher performance than some home or personal computers that are selling today for up to $2000. It is more powerful than computer systems which cost over $20,000 in the early 1970's.

This new machine can entertain your whole family with spectacular video games and cartoons, made possible by its ultra high resolution graphics and fast BASIC. It can help you with your personal finances and budget planning, made possible by its decimal arithmetic ability and cassette data storage capabilities. It can assist you in school or industry as an ultra powerful scientific calculator, made possible by its advanced scientific math functions and built-in "immediate" mode which allows complex problem solving without programming! This computer can actually entertain your children while it educates them in topics ranging from naming the President of the United States to tutoring trigonometry all possible by its fast extended BASIC graphics and data storage ability.

The machine can be economically expanded to assist in your business, remotely control your home, communicate with other computers and perform many of the other tasks via the broadest lines of expansion accessories in the microcomputer industry.

This machine is super easy to use because it communicates naturally in BASIC, an English-like programming language. So you can easily instruct it or program it to do whatever you want, *but you don't have to*. You don't because it comes with a complete software library on cassette including programmes for each application stated above. Ohio Scientific also offers you hundreds of inexpensive programs on ready-to-run cassettes. Program it yourself or just enjoy it, the choice is yours.

**Standard Features**

- Uses the ultra powerful 6502 microprocessor
- 8K Microsoft BASIC-in-ROM
- Full feature BASIC runs faster than currently available personal computers and all 8080-based business computesr.
- 4K static RAM on board expandable to 8K
- Full 53-key keyboard with upper-lower case and user programmability
- Kansas City standard audio cassette interface for high reliability
- Full machine code monitor and I/O utilities in ROM
- Direct access video display has 1K of dedicated memory (besides 4K user memory), features upper case, lower case, graphics and gaming characters for an effective screen resolution of up to 256 by 256 points. Normal TV's with overscan display about 24 rows of 24 characters, without overscan up to 30 x 30 characters.

**Extras**

- Available expander board features 24K static RAM (additional mini-floppy interface, port adapter for printer and modem and OSI 48 line expansion interface.
- Assembler/editor and extended machine code monitor available.

Commands

| | | | | | |
|---|---|---|---|---|---|
| CONT | LIST | NEW | NULL | RUN | |

Statements

| | | | | | |
|---|---|---|---|---|---|
| CLEAR | DATA | DEF | DIM | END | FOR |
| GOTO | GOSUB | IF...GOTO | IF...THEN | INPUT | LET |
| NEXT | ON...GOTO | ON...GOSUB | POKE | PRINT | READ |
| REM | RESTORE | RETURN | STOP | | |

Expressions

Operators

—, +, *, /, ↑, NOT, AND, OR, >, <, <>, >=, <=, =

RANGE $10^{-32}$ to $10^{+32}$

Functions

| | | | | | |
|---|---|---|---|---|---|
| ABS(X) | ATN(X) | COS(X) | EXP(X) | FRE(X) | INT(X) |
| LOG(X) | PEEK(I) | POS(I) | RND(X) | SGN(X) | SIN(X) |
| SPC(I) | SQR(X) | TAB(I) | TAN(X) | USR(I) | |

String Functions

| | | | | | |
|---|---|---|---|---|---|
| ASC(X$) | CHR$(I) | FRE(X$) | LEFT$(X$,I) | LEN(X$) | MID$ (X$,I,J) |
| RIGHT$(X$,I) | | | STR$(X) | | VAL(X$) |

Plus variables, arrays and good editing facilities.

Fully built and tested. Requires only +5V at 3 amps and a videomonitor or TV and RF converter to be up and running.

At this price there is going to be an enormous demand for Superboard II and supplies are going to be extremely limited. Lotus Sound is expecting first deliveries from the States in mid-December and orders will be treated strictly in the order they arrive, so post off today if you want to avoid long delays.

**To: LOTUS SOUND**
**4 MORGAN ST., LONDON E3 5AB**

Please send me ..... Ohio Scientific Superboard Computer(s)
I enclose cheque/PO for £ ................
Name ................
Address ................
................
................
................

# SOUND

**4 MORGAN ST., LONDON E3 5AB**
(Mail Order only)

# NewBear Computing Store

## ★ SYM-1 ★

(Formally VIM-1)

The new 6502 micro from Synertek. Fully assembled and tested

Ex Stock @ **£199.00**
+8% V.A.T.
Carriage £1.00

## ★ NASCOM 1 ★

**Z80 BASED MICROCOMPUTER KIT**

Ex Stock @ **£197.50**
+ 8% VAT
£1.00. Carriage

## ★ SALE ★

FOR CALLERS ONLY AT BONE LANE, NEWBURY. 1 ONLY V.D.U. £300 + 8% V.A.T. CASH ONLY.

A LARGE NUMBER OF FULLY POPULATED LOGIC BOARDS, EDGE CONNECTORS, P.C.B.s AND COMPONENTS ALL AT SILLY PRICES. PLUS SPECIAL OFFERS ON CERTAIN I.C.s.

CAN YOU BEAR TO MISS THIS OPPORTUNITY? SAT., 6th JAN. TO SAT., 13th JAN. INCLUSIVE.

| | |
|---|---|
| Z80 CPU 2.5 MHz | £13.99 |
| Z80 P10 PS | £9.15 |
| Z80 CTC PC | £9.15 |
| Z80A CPU 4MHz | £20.00 |
| Z80A P10 | £11.05 |
| Z80A CTC | £11.05 |
| MC 6800 | £8.75 |
| MC 6820 | £4.63 |
| MC 6850 | £6.74 |
| MC 6810 | £3.61 |
| MC 8602P | £2.88 |
| MC14536P | £3.69 |
| MC 3459 | £2.53 |
| SC/MP II | £10.30 |
| 6502 | £14.93 |
| 8080A | £8.00 |
| **Memories** | |
| 2102-1 | £1.25 |
| 2102L-1 | £1.35 |
| 2112 | £3.04 |
| 2513 | £6.50 |
| SWATBUG | £16.00 |
| 2114 | £8.25 |

| | |
|---|---|
| 2708 | £6.99 |
| 2716 | £22.50 |
| MIKBUG | £13.65 |
| 4116 | £15.00 |
| **Interfacing I.C.'s** | |
| SFF 96364 | £16.20 |
| MC 1488P | £1.40 |
| MC 1489P | £1.40 |
| 75150P | £1.30 |
| 75150N | £1.20 |
| 75154 | £2.30 |
| 4N33 | £1.95 |
| AY-5-1013 | £4.50 |
| 6402 | £4.50 |
| 6571 | £6.70 |
| 2501 | £6.50 |
| **Buffers** | |
| 81LS95 | £1.30 |
| 81LS96 | £1.30 |
| 81LS97 | £1.30 |
| 81LS98 | £1.30 |
| 8T26 | £1.84 |
| 8T95 | £1.60 |
| 8T97 | £1.60 |
| 74367 | £1.30 |

### 24-HOUR TURNROUND ON ORDERS

Please add 8% VAT to all prices. P&P 30p unless otherwise stated. Barclaycard and Access welcome. Overseas orders issued with Pro-Forma Invoice. Send for catalogue to Newbear Computing Store, Bone Lane, Newbury, Berks. Callers welcome Mon.-Sat., 9.00-5.30, but please phone us first on 0635 49223. New office — 2 Gatley Road, Cheadle, Cheshire. (Callers only). Telephone 061-491 0134.

## SEND FOR OUR NEW CATALOGUE

| | | P&PP |
|---|---|---|
| **COMPUTER DESIGNS** | | |
| 77-68 6800 Microcomputer | £7.50 | .50 |
| WB-1 TTL Microcomputer | £6.50 | .50 |
| **FROM ADAM OSBOURNE ASSOCIATES** | | |
| Introduction to Microcomputers | | |
| Volume 0 The Beginners Book | £5.95 | .50 |
| Volume 2 Basic Concepts | £5.95 | .50 |
| Volume 2 Some Real Products June 1977 Revision | £11.95 | 1.00 |
| 8080A/8085 Assembly Language Programming | £6.95 | .50 |
| 6800 Assembly Language Programming | £6.95 | .50 |
| Some Common Basic Programmes | £5.95 | .50 |
| 6800 Programming for Logic Design | £5.95 | .50 |
| 8080 Programming for Logic Design | £5.95 | .50 |
| Payroll with Cost Accounting in Basic | £9.95 | 1.00 |
| Instant Freeze-Dried Computer Programming in basic. By Jerald R. Brown | £4.95 | .75 |
| My Computer Likes Me When I Speak in Basic | £1.65 | .30 |
| Computer Programs that Work | £2.40 | .75 |
| Basic Software Library Volume 1 | £17.50 | .50 |
| Volume 2 | £17.50 | .50 |
| Volume 3 | £26.95 | .50 |
| Volume 4 | £7.95 | .50 |
| Volume 5 | £7.95 | .50 |
| Volume 6 | T.B.A. | .50 |
| Volume 7 | £26.95 | .50 |

### NEW BOOKS

| | | |
|---|---|---|
| First Book of Kim | £7.00 | .50 |
| Z80 Microcomputer Handbook | £7.25 | .50 |
| Using the 6800 Mircoprocessor | £6.25 | .50 |
| Micro 6502 Journal | £1.70 | .50 |
| Learning Basic Fast | £6.30 | .50 |
| How to Program Microcomputers | £6.30 | .75 |
| Getting involved with your Own Computer | £4.75 | .75 |
| 8080A Bugbook | £6.95 | .75 |
| **SYBEX** | | |
| Microprocessors C201 | £8.00 | .50 |
| Microprocessor Interfacing Techniques | £8.00 | .50 |

### BEAR BAGS

1. 77-68 CPU PCB and Components £49.50
2. 77-68 LED's and Switches £14.95
3. 77-68 Power Supply £17.95
4. 77-68 19 5u Rack and Backplane £26.70+
5. 77-68 4K Ram PCB and Components £74.00
6. 77-68 Mon 1 PCB and Components £50.00
7. 4K Ram Exorciser PCB and Components £71.50
8. 8K Ram Exorciser PCB and Components £160.00*
9. Petitevid VDU Kit £85.00
10. Kansas City Cassette Interface £18.95
11. UHF Modulator £4.50
12. 77-68 VDU PCB and Components £69.50
13. 77-68 Mon 2 PCB and Components £t.b.a.
14. Prom Programmer PCB and Components £35.00
15. PROMVERTER (Enables a 2708 to be used instead of MIKBUG) £8.50

50p Postage and Packing unless otherwise stated + £1.50 Postage and Packing.
*£1.00 Postage and Packing.

### FLOPPY DISC DRIVES

5½" SHUGART SA400 DISC DRIVE £255.00
Carriage by Securicor 'C' Rate . . . £4.50

8" DRI SINGLE SIDED DOUBLE DENSITY . . . £325.00
Carriage by Securicor 'C' Rate . . . £5.00

8" DRI DOUBLE SIDED DOUBLE DENSITY . . . £365.00
Carriage by Securicor 'C' Rate . . . £5.00